Python » MySQL » Slashdot » OpenMoko

LINUX FORMAT

Главное в мире Linux

№ 10 (97)
Октябрь 2007

LXF DVD

MOPSLinux 6.0

ПЛЮС: ALT Linux 4.0 Personal Desktop

» Zenwalk Live & Core » KDE 4 Beta 2



Борьба за лучший ubuntu

Эксклюзивная информация о том, как делается самый популярный в мире дистрибутив

» Что нового в Gusty Gibbon?

» Мнения разработчиков

» ПЛЮС! Собери свой собственный Ubuntu

Не заSlashdot'ишь! Подготовьте свой web-сайт к наплыву посетителей **с. 42**

Bash, Sed, Awk Накоротке с основными инструментами командной строки **с. 58**

Ebuild'нем... Приготовьте исходный код для пользователей Gentoo **с. 72**

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974



«Я ни на секунду не предлагаю OOo стать Боргом!»

OpenOffice.org Луис Суарес-Поттс

Главное в мире Linux

К Вашим услугам...

Продав компанию Thawte, основатель Ubuntu Марк Шаттлворт заработал 575 миллионов долларов.
А на что бы вы потратили эту сумму, окажись она в вашем распоряжении?



Пол Хадсон
Я бы создал фонд помощи бездомным детям... а затем купил большую, блестящую машину.



Грэм Моррисон
Я бы притворился бездомным ребенком, чтобы урвать еще и кусок, доставшийся Полу.



Майк Сондерс
Я бы проспонсировал строительство механических бездомных детей, чтобы они делали работу, которую сейчас выполняют живые бездомные дети.



Эфрейн Эрнандес-Мендоса
Я бы не сказал об этом ни одной живой душе, а сам продолжил жить по-старому: переведя доллары в стерлинги, вы получите не такую уж впечатляющую сумму.



Мэтт Нейлон
Я бы вложил один-единственный пфенниг под 5% с капитализацией, подождал 2000 лет и получил бы сумму, в 134 млрд. раз превышающую золотой запас Земли.



Энди Ченнел
Я бы закупил множество танков, самолетов, винтовок и тонны боеприпасов для своих друзей, сражающихся в армии CIRCA. www.clownarmy.org



Энди Хадсон
А кто-нибудь уже оплатил перевод Linux на эсперанто или классическую латынь? А то я бы, думаю, взялся...



Д-р Крис Браун
Я бы подарил ноутбук с Ubuntu каждому сотруднику Microsoft, чтобы просто посмотреть на их лица.



Нейл Ботвик
Я бы, пожалуй, засунул большую часть этого добра за спинку дивана – по крайней мере, до сих пор мои деньги собирались именно там.

Миссия журнала

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Corporate Patron Free Software Foundation



В плену у технологий

Жить без технологий невозможно – по крайней мере, если вы с удовольствием читаете журналы вроде LXF. Компьютеры (как проявление технологий) берут на себя тяжелую и рутинную работу вроде размещения на бумаге символов, которые я сейчас набираю – а вам остается только пожинать плоды заслуженного бездействия... пардон, интеллектуальной деятельности. Но не кажется ли вам иногда, что технологий стало слишком много?

Достаточно интересными в этом смысле являются настольные поисковые системы наподобие Google Desktop. Вырастающие корнями из мощных движков, способных выловить крупицы полезной информации из безбрежного океана хлама, известного как World Wide Web, они способны за секунду отыскать нужный документ и на вашем локальном компьютере... если, конечно, он уже представляет собой все тот же океан или хотя бы пруд хлама!

Поймите меня правильно – я не люмпен и вполне способен оценить технические достоинства подобных инструментов, и даже вижу несколько случаев, когда они могут оказаться действительно полезными. Но не будет ли правильнее побороть собственную неаккуратность, а не сваливать эту проблему на мощный, но безмозглый компьютер? По крайней мере, до тех пор, пока он не научится прибираться и в вашей комнате тоже... LXF

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru

› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.128

LINUX FORMAT

Октябрь 2007 №10 (97)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники



LXF DVD97



MOPSLinux 6.0
Последняя версия отечественного дистрибутива, базирующегося на Slackware Linux 12.0 – большой выбор пакетов и очень качественная русификация с сохранением достоинств оригинала!

Zenwalk 4.6 Live & Core
Современная, быстрая, рациональная разновидность Slackware, запускающаяся прямо с CD. А также неплохой фундамент для собственной разработки.

ALT Linux 4.0 Personal Desktop
Настольный дистрибутив от российской команды разработчиков, основанный на одном из крупнейших в мире репозиториев свободного ПО – Sisyphus.

Подборка Сравнений
Убойная коллекция обзоров из предыдущих выпусков нашего журнала поможет найти подходящее ПО для работы и отдыха.



› Выиграйте пару секунд на этом круге в SuperTuxCart.

Что за штука... с. 46

Open Moko?

Закажите себе комплект разработчика и отправьте разрекламированный iPhone на свалку истории!





LXF HotPicks



› Thunder and Lighting: обрушьте на врага гром и молнии!

LXF ИНТЕРВЬЮ

«Я ни на секунду не предлагаю OOo стать боргом...»

Луис Суарес-Поттс с. 22





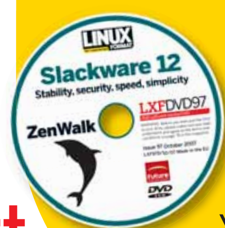
Переживем слэш-доттинг с. 42

Помогите своему web-серверу выстоять, когда в его двери начнут ломиться тысячи посетителей.



Подпишись на Linux Format и сэкономь!

Содержание



LXF DVD внутри!

Хотите узнать, что на нем? См. с. 120

Спецрепортаж

Борьба за лучший Ubuntu

Фаворит 2007 года в объективе нашего микроскопа с. 26



А также...

Постоянные рубрики



› **Перекодировать видео из DVD в открытый формат – это просто!**

Обзоры



› **Хорошие или плохие – но они поставляются с Linux из коробки.**

Сравнение: IRC-клиенты



LINUX FORMAT

Новости

ГЛАВНЫЕ НОВОСТИ: » Дистрибутив для школ России » OOXML отклонен » AMD открывает спецификации ATI » Игрушечный прогресс » Франция выбирает Red Hat » Проблемы заморского ГУИНа » Открытый QNX

» Рубрику ведет Илья Шпаньков





БИТВА ЗА ОБРАЗОВАНИЕ: раунд первый

Решение российского правительства параллельно с закупкой проприетарного ПО для школ создавать и альтернативные проекты на базе свободного ПО стало логическим завершением многочисленных споров и обсуждений за круглыми столами перспектив дальнейшего развития отечественного программного обеспечения. Проект, подготовленный Мининформсвязи, подразумевает почти трехлетнюю программу создания школьного Linux-дистрибутива, способного полностью заменить закупаемое сегодня коммерческое ПО: в конце текущего года будет проведен тендер, победитель которого в 2008 году осуществит пилотные переводы на свободное ПО трех регионов, в списке которых сегодня звучат Татарстан, Пермский край и Томская область. В течение тестового периода будут вноситься все необходимые корректировки, а к осени 2009 года планируется обеспечить операционной системой Linux все российские школы. Финансирование столь глобального проекта будет осуществляться из федерального бюджета, при этом в 2007 году будет освоено 50 млн. рублей, в 2008 году – 60 млн., а в 2009 году сумма затрат составит около 600 млн. рублей.

Масштабность поставленных правительством задач стала причиной объединений российских компаний, работающих в области свободного ПО, сразу в несколько мощных структур. Первыми заявили о себе участники альянса, организованного компанией ALT Linux. К ней присоединились такие известные разработчики и внедренцы, как Линукс-онлайн, Naumen, Eterrsoft и системный интегратор «Центр». Основной задачей данного объединения станет разработка специализированного дистрибутива Linux и создание разветвленной сети специализированных филиалов, которые будут осуществлять техническую поддержку и сопровождение дистрибутива в учебных заведениях. Школьный Linux будет создаваться на базе ALT Linux Junior и репозитория Sysiphus, являющегося третьим в мире по числу входящих в него пакетов. При этом основной упор участники альянса делают на то, что созданием школьной операционной системы должны заниматься разработчики, обладающие собственной пакетной базой.

Спустя всего несколько дней заявил о себе и второй альянс. Его инициатором выступила петербуржская компания Линукс Инк, а свой интерес к данному объединению проявили ЗАО Lynx, Линуксцентр, Инфра-Ресурс, издательство BHV и НПО «Сеть». Программа данного альянса немного отличается от той, что озвучили конкуренты: не зацикливаясь на каком-то определенном дистрибутиве, разработчики планируют заняться созданием мультисистемного специализированного программного обеспечения по трем направлениям: педагогическое и обучающее ПО, затем – блок, обслуживающий деятельность школы как хозяйствующего субъекта, и, наконец, блок программ, обеспечивающих деятельность школы как социального учреждения. В целом данный вариант выглядит более взвешенным, ведь на самом деле не имеет смысла избавляться от одного монополиста, чтобы заменить его другим. Школы должны иметь свободу выбора и пользоваться той операционной системой, которая будет наиболее выгодна и удобна в работе, а задача разработчиков – подготовить качественное программное обеспечение, способное функционировать в любой среде.

Двоевластия пока не будет

В начале сентября технический комитет Международной организации по стандартизации (ISO) вынес свое решение – формат OOXML не собрал количество голосов, необходимых для выдвижения его в качестве международного стандарта одновременно с ODF. Такой результат был получен на основании решений организаций по стандартизации из более чем ста стран. По правилам, для выдвижения в число стандартов «кандидат» должен набрать не менее 2/3 голосов «за» и не более 25% «против». Ни одно из этих условий не было достигнуто: число сторонников OOXML составило лишь 53% от общего количества проголосовавших, при этом против нового стандарта высказалось около 26%. Но такой результат еще не означает, что у Microsoft нет шансов протолкнуть собственную разработку: в феврале будущего года пройдет повторное заседание технического комитета ISO и до этого времени редмондский гигант может исправить недочеты, обнаруженные экспертами в спецификациях нового формата, занимающих около 6000 страниц. А до этого времени есть повод разобраться с тем, как проходило голосование в региональных комитетах. И повод этот довольно серьезный: в некоторых странах рассмотрение формата OOXML сопровождалось массой нарушений действующего законодательства. Сторонние наблюдатели постоянно сообщали о различных подтасовках, попытках ввести участников комиссии в заблуждение или просто попытках победить числом, а не уменьем: были зафиксированы случаи, когда состав технической комиссии в последние часы голосования резко возрастал за счет вхождения в него компаний, являющихся партнерами Microsoft. В Швеции вообще пришлось аннулировать результаты голосования по причине явных подтасовок.

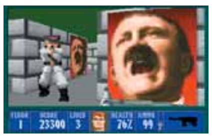
Не лучшим образом себя показали и страны СНГ, единодушно поддержавшие инициативу Microsoft по продвижению OOXML. Причем, то, что Россия без каких-либо замечаний одобрила OOXML, послужило причиной появления официального запроса в Федеральное агентство по техническому регулированию и метрологии, подготовленного депутатом Госдумы РФ Виктором Алкснисом. Согласно просочившимся в прессу данным, решение об одобрении нового формата было вынесено Федеральным агентством без сколько-нибудь серьезных исследований спецификаций, а непосредственными исполнителями стали рядовые сотрудники учреждения. В связи с этим депутат попросил руководство агентства сообщить, каким образом принималось решение, кто проводил техническую оценку представленных спецификаций, какие экспертные группы привлекались к работе комиссии и кто конкретно вынес решение об одобрении формата OOXML. Одновременно он запросил копии всех решений, протоколов и заключений специалистов, если они, конечно, имеются.

Дела игрушечные



Все продолжающийся рост популярности свободных операционных систем оказывает влияние на весь рынок программного обеспечения, но, пожалуй, наибольший интерес у пользователей вызывает ситуация с развитием игровых приложений для Linux. А за прошедший месяц здесь произошло немало интересных событий. Впрочем, начать обзор игровых новостей придется с не самого оптимистичного сюжета: компания id Software решила поменять приоритеты и готовит новую игру Rage с движком, использующим только DirectX, без поддержки OpenGL. Следует отметить, что раньше компания довольно благосклонно относилась к Linux-геймерам: достаточно вспомнить, что такие популярные игры, как Quake и Wolfenstein после определенного периода коммерческого распространения передавались сообществу Linux-пользователей, а движок легендарного Doom III изначально создавался в платформенно-независимом варианте. Не последнюю роль в подобной универсальности играл тот факт, что движки всех вышеперечисленных игр были построены на платформе OpenGL, что значительно облегчало перенос игровых приложений в среду Linux. Чем вызван переход на более закрытый DirectX, руководство компании id Software умалчивает.

Впрочем, на рынке немало и таких разработчиков, кто наоборот пытается успеть завоевать в Linux-сообществе свою долю пользователей. Компания CCP Games совместно с TransGaming Inc. планируют к концу текущего года разработать клиентскую часть для популярной онлайновой игры EVE Online не только для Windows, но и для платформ Linux и Mac OS X. Ранее фанаты данной космической RPG пользовались клиентской частью для Windows, запуская ее с помощью пакета Cedega. Хороший опыт не остался незамеченным и теперь авторы Cedega смогут избавить своих пользователей от дополнительных забот. Аналогичное решение приняла и компания Hothead Games, создающая на базе онлайновой версии Penny Arcade мультиплатформное игровое приложение. Объясняют свой поступок авторы игры очень просто: рынок Linux-систем стремительно растет и если сейчас сделать правильные шаги, то в недалеком будущем можно будет значительно расширить число своих пользователей за счет приверженцев свободного ПО. Но, пожалуй, наиболее радужные перспективы ожидают тех разработчиков, которые не останавливаются на полпути и переводят свои игровые приложения под свободные лицензии. Один из таких примеров – компания Viewizard, выпустившая свой космический 3D-шутер AstroMenace сразу под двумя лицензиями: коммерческой и свободной GPLv3. Благодаря такому подходу разработчики не только наращивают число своих пользователей, но и привлекают талантливых разработчиков к дальнейшему совершенствованию игры.



Новости короткой строкой

» Компания Microsoft закрыла web-сайт «Get The Facts», на котором публиковались специальным образом подготовленные доказательства преимуществ Windows перед Linux.

» Компания ALT Linux объявила о готовности долгожданной версии ALT Linux Personal Desktop 4.0.

» Компания LinuxCenter по многочисленным просьбам приступила к продажам электронной версии журнала Linux Format.

» В интернете открылся web-сайт Linux Wikipedia, на котором планируется собрать рекордное количество документации.

» Компания IBM присоединилась к числу разработчиков пакета офисных программ OpenOffice.org.

» 5 сентября стартовал глобальный образовательный проект SELF Platform, призванный объединить усилия преподавателей из разных стран по созданию документации по Linux.



Фото: http://www.bigfoto.com

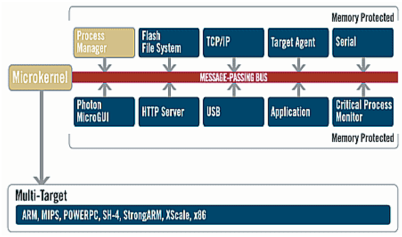
Созданы друг для друга

Пока российские власти только планируют осторожное внедрение Linux в сферу образования, в Европе миграционные процессы набирают обороты. Министерство образования Франции и компания Red Hat сообщили о завершении процесса перевода 2500 серверов, установленных в 30 подразделениях министерства по всей стране, на использование операционных систем Red Hat Enterprise Linux. Основной причиной столь масштабной миграции послужила государственная программа по уменьшению затрат на содержание правительственных учреждений Франции, а также необходимость в снижении зависимости от определенных поставщиков проприетарного ПО. По словам чиновников, сегодня практически весь парк серверов министерства работает только на Linux, а выбор в пользу Red Hat был обусловлен тем, что данная компания обеспечивает полный спектр услуг, включая техническую поддержку и сопровождение своих программных продуктов.

Еще более значимые результаты достигнуты в Германии, где власти федеральной земли Северный Рейн-Вестфалия заключили с компанией Novell контракт на обеспечение 33 университетов, насчитывающих около 560 тысяч студентов, операционными системами SUSE Linux. Примечательно, что устанавливаться данное ПО будет не только на рабочие машины в аудиториях, но и на персональные компьютеры студентов и работников университетов. Следует также отметить, что чуть раньше аналогичные контракты были реализованы в Тюрингии и Баварии, где число студентов составляет примерно 300 тысяч человек. Таким образом, по окончании миграции в общей сложности почти 40% всех германских студентов будут пользоваться ОС Linux.

Свободный Neutrino

Компания QNX Software Systems объявила об открытии кода операционной системы реального времени QNX Neutrino RTOS. Теперь данная разработка будет выходить под двойной лицензией, позволяющей как принимать участие в совершенствовании данного ПО свободным разработчикам, так и распространять её на прежней коммерческой основе. Таким образом авторы проекта надеются создать сообщество разработчиков открытого ПО, принимающих полноправное участие в развитии программного продукта. На первом этапе в разряд открытого ПО перешли исходные тексты микроядра QNX Neutrino, а также библиотеки C и некоторых драйверов. В дальнейшем ожидается открытие кода Photon microGUI, сетевого стека, файловой системы и набора утилит. Специально для сторонних разработчиков авторы проекта подготовили специализированный web-сайт Foundry27, после регистрации на котором можно будет получить доступ к исходным текстам QNX Neutrino RTOS.



AMD открывает спецификации драйверов ATI



Сентябрь принес радостную новость для всех пользователей операционных систем Linux, обладающих видеокартами с лейблом ATI: компания AMD решила открыть спецификации этих графических чипсетов. Одновременно с выпуском очередной версии драйверов компания начала выкладывать в свободный доступ технические материалы, описывающие все тонкости функционирования видеокарт. На сегодняшний день в запасниках AMD находится более 72 тысяч файлов с документацией и ожидается, что большая часть из них будет постепенно представлена на специализированном web-сайте, открытом по такому случаю компанией AMD. Данное решение должно способствовать ускорению разработки свободной версии драйверов, а также позволит компании привлечь к созданию более качественных версий ПО сторонних разработчиков. Также руководство AMD пообещало всячески помогать сообществу в создании свободных видеодрайверов для карт ATI. И, похоже, надежды компании на улучшение ситуации с полноценным 3D-ускорением в Linux имеют под собой почву: очередная версия драйверов на тестах показала значительный прирост в скорости работы, поэтому есть надежда, что в скором времени все мучения владельцев видеокарт ATI останутся в прошлом.

Осуждение с принуждением



Слишком широкое распространение проприетарного ПО создает, порой, самые непредсказуемые проблемы, и с одним из таких примеров столкнулся бывший администратор одного из трекеров EliteTorrents BitTorrent Скотт МакКосланд [Scott McCausland]. В 2006 году он выложил на сервер пиратскую копию очередной серии кинохита «Звездные войны», за что был арестован и осужден на полгода тюремного заключения и еще полгода домашнего ареста. Его история осталась бы незамеченной, как и тысячи других, но случилось так, что Скотт оказался приверженцем свободного ПО и предпочитал работать на компьютере исключительно в операционной системе Ubuntu. Между тем, надзорные органы по закону обязаны установить на компьютер осужденного специализированное ПО, отслеживающее активность пользователя в сети и ежедневно отсылающее отчеты в управление по надзору за осужденными. Как нетрудно догадаться, данное программное обеспечение было предназначено для установки исключительно в операционную систему Windows. Впрочем, офицеры быстро решили проблему, предложив Скотту удалить Ubuntu и установить «правильное» ПО, на что убежденный сторонник свободного ПО не мог согласиться ни под каким предлогом. Посовещавшись с адвокатом, он решил подать в суд и потребовать, чтобы надзорные органы предоставили такое программное обеспечение, которое смогло бы работать и в Ubuntu Linux. При всей курьезности ситуации на самом деле она приоткрывает еще один серьезный аспект неблагоприятного влияния проприетарного ПО на жизнь общества: при определенных обстоятельствах оно может способствовать нарушению основных прав и свобод, гарантированных конституцией.

» При подготовке данного выпуска новостей были использованы были использованы материалы сайтов: Приоритетные национальные проекты, InfoWorld, vnunet.com, ars technica, Free Software Foundation, Citrix и других, а также наших собственных корреспондентов.

LINUX FORMAT Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Его слабости – mass storage, разметка диска и файловые системы.

ZFS: конец файловым проблемам?

Разметка диска и файловые системы – те немногие элементы «высшей математики» POSIX'ивизма, представление о которых не помешает любому пользователю. Причем получить его желательно до установки первого дистрибутива. Иначе трудно исключить ситуацию, когда будет мучительно больно за бесцельно потраченное дисковое пространство в одной ветке файлового древа и острую нехватку оного – в другой.

Конечно, методы перераспределения дискового пространства между разделами существуют. Это и программы типа parted, и механизмы логических томов (например, LVM в Linux). Однако использование их не вполне тривиально, да и не всегда безопасно для данных.

ZFS, похоже, предлагает окончательное решение этого вопроса. Эта 128-битная (!) файловая система была разработана фирмой Sun для ОС Solaris. Потом она была включена в OpenSolaris, портирована во FreeBSD и (через FUSE) поддержана Linux. Ожидается, что она будет файловой системой по умолчанию в Mac OS X Leopard.

В ZFS наличное дисковое пространство предстает в виде единого пула (zpool), доступ к которому могут иметь все включенные в него ветки файловой иерархии. Кроме того, 128 разрядов ZFS гарантируют, что с ограничениями на ее размер не придется столкнуться за всю грядущую историю человечества. По словам создателя XFS Джеффа Бонвика [Jeff Bonwick], «для этого пришлось бы вскипятить океан». Наконец, целостность данных и быстродействие при доступе к ним – тоже не последние козыри этой ФС.

Похоже, буква Z в названии действительно символизирует... точку в развитии файловых систем.

alv@posix.ru

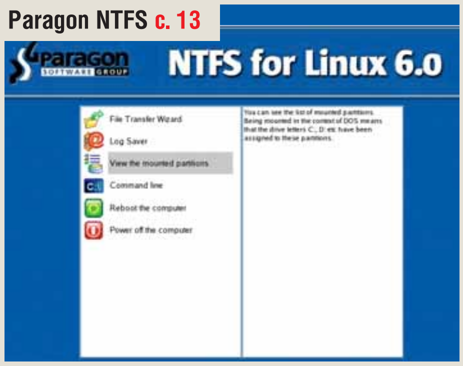
Сегодня мы рассматриваем...

eSys ePC с. 08



› **Имеет смысл потратить на этот компьютер пару сотен – и потом держать в качестве постоянно включенного сервера или второго рабочего места.**

Paragon NTFS с. 13



› **Если вы – системный администратор или просто имеете машины Windows в домашней сети, это бывает удобным.**

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth
Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

› *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.*

Рейтинг 9/10

eSys ePC basic

Данный ПК – еще не ноутбук за $100, но он доказывает, что концепция реализуема. **Ник Вейч** проверяет, может ли такая удача быть правдой…

Система

» Процессор: Celeron 331 (2,66 ГГц, кэш 256 КБ, системная шина 533 МГц)
» RAM: 512 МБ
» Накопители: жесткий диск 80 ГБ, DVD-ROM
» Графика: VIA Unichrome (работает в Vesa-режиме)

Ну что можно купить за 280 долларов? Кусок PlayStation 3? 736 консервных банок фасоли? Четыре подписки на лучший в мире журнал о Linux? А настольный компьютер с предустановленным Linux – не хотите? Компания eSys – известный производитель систем, которые специалисты по маркетингу называют «доступными», в диапазоне от самых простых до умеренно «навороченных». Наша модель – простейшая из простых – дешева до предела. Но: дешевка ли?

Качество изготовления – хоть куда. Чашку кофе можно ставить сверху без опаски. И хотя корпус – гадкий midi-tower серебристого пластика от магазина бытовой техники High Street, свое дело он делает. Предусмотрены свободные отсеки для дополнительного оборудования (отрада самоделкиных), несколько USB- и аудио-портов спереди и обычный часток выводов сзади, включая стандартный VGA-разъем для монитора.

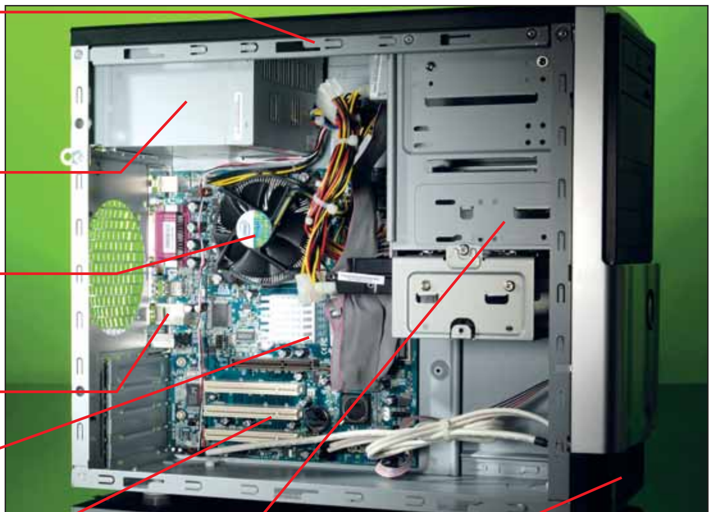
В комплект входят мультимедиа-клавиатура и мышь, а вот на монитор придется раскошелиться. Найдите такой, чтобы работал со стандартным VGA-входом – впрочем, многие ЖК-дисплеи это до сих пор умеют.



› Чтобы максимально снизить цену, ePC продается без монитора.

По части графики, материнская плата, на которой собрана система, оснащена встроенным графическим чипсетом VIA, но это не есть хорошо. Конечно, VIA позаботилась о поддержке Linux на своем оборудовании, но лучший результат зачастую достигается при использовании свободных драйверов, не полностью поддержанных VIA. Однако даже проект Openchrome, всеобщая палочка-выручалочка, на время написания статьи не поддерживал данный чипсет. Винить eSys особо не за что, но, честно говоря, компания могла бы подобрать материнскую плату от Intel с интегрированной поддержкой графики. Досадно осознавать, что потенциал аппаратуры используется не полностью. Стоит, правда, заметить, что улучшенная поддержка графики могла бы повлиять разве что на качество воспроизведения фильмов – того, кто ищет мощную графическую станцию, наша машина вряд ли заинтересует.

ePC изнутри



Накопители
Материнская плата оснащена поддержкой SATA, но и DVD-привод, и 80-ГБ жесткий диск являются PATA-устройствами.

Питание
250-ваттный блок питания работает тихо, но безотказно.

Процессор
Под мощным кулером скрывается 2,66-ГГц Celeron.

Обмен информацией
Единственный сетевой порт 10/100.

Графика
«Бортовая» графика VIA Unichrome прячется здесь.

Слоты расширения
3 слота PCI и один AGP дают шанс на разгон.

Резервы роста
3 отсека спереди и 1 внутри предусмотрены про запас.

Периферия
Два USB-порта, расположенные на передней панели, удобны для подключения цифровых фотоаппаратов и т.п.

Ubuntu навсегда

В комплект входят клавиатура и мышь. Оба устройства оснащены разъемами PS/2, и это радует. Клавиатура выглядит вполне солидно, есть все функциональные клавиши, а сверху еще и панель медиакнопок. Удивительно, но все они (ну, почти все) исправно работают! Разве не здорово, что можно изменить громкость или запустить браузер без мороки с X-настройками или lineak?

Менее удивительно то, что сборщики eSys EPC выбрали в качестве системы для установки Ubuntu (между прочим, компьютер снабжа-



› Интересно было бы в очередном номере сравнить предустановленный Linux в исполнении Dell.

ется сопроводительным диском с драйверами VIA, хотя толку от него мало, ведь на нем нет специальных Linux-драйверов). Это уже тенденция, и с учетом того, что Dell тоже собирается ставить Ubuntu, выбор явно неплох. Не могут же ошибаться сразу несколько миллионов пользователей, верно?

Разочаровывает, однако, что это голая инсталляция Ubuntu. Похоже, произошло следующее: Ubuntu (речь идет о 6.06 LTS) установили на компьютер, использовав стандартное имя пользователя и пароль (кстати, они есть на стикере на боку коробки!), затем полученный образ залили на жесткие диски всех новых компьютеров. Зная это, нетрудно предугадать последствия подобной установки Ubuntu.

Например, если вы задумали подарить такой компьютер племяннику на Рождество, готовьтесь к продолжительным телефонным переговорам. После первой загрузки и входа в систему пользователь получает стандартный, пустой рабочий стол Ubuntu: ни на экране, ни в коробке нет никаких сведений о том, что делать дальше. Невольно возникает идея сразу же сменить пароль и/или настроить новую учетную запись!

Но это не единственная проблема. Конечно, бывалый линуксоид запросто добьется правильной работы, установив дополнительные пакеты, но ведь большинство покупателей будут новичками. Так вот, когда они запустят *Firefox* и набредут на какой-нибудь сайт с Flash-эффектами, обнаружится, что по факту ничего не работает. *Firefox*, естественно, предложит загрузить подходящий модуль, но мы-то с вами знаем, что из этого ничего не выйдет; а пользователь зайдет в тупик. Инсталлировать Flash-player от Adobe вручную для новобранца весьма непросто, даже если

> «Неудивительно, что производители выбрали дистрибутив Ubuntu.»

он знает, откуда его можно скачать. Политика Adobe в отношении OEM настолько либеральна, что производители eSys могли бы преспокойно предустановить Flash. Пусть такое ПО может «запятнать» систему щепоткой «несвободы», но лица, кому эти пятна не дают покоя, обычно способны инсталлировать и стерильную систему с сопроводительного диска.

Выходит, что это не проблемы Ubuntu. Да это и не проблемы самого устройства: скорее, упущенные возможности завоевать симпатии пользователей, что могло бы благотворно отразиться на имидже Linux.

Компьютер явно заслуживает внимания; его легко использовать как домашний сервер или настольную систему для непритязательного пользователя. Скорости процессора вполне достаточно для повседневных нужд. Правда, воспроизведение фильмов несколько отрывистое, а 3D-эффекты Compiz недоступны, но ведь для рабочей лошадки это не главное. Зато устройство вызывающе дешево, и в этом есть своя «изюминка». Мы рекомендуем взять да и установить на купленный компьютер свой любимый дистрибутив – терять особо нечего. LXF

LINUX FORMAT Вердикт

eSys ePC basic

Разработчик: eSys
Сайт: www.esysglobal.com
Цена: примерно $280 с НДС

Функциональность	6/10
Производительность	7/10
Простота использования	5/10
Цена	10/10

» *Купи один или парочку! Подари близким! Да не забудь отключить телефон, а то замучают техническими вопросами…*

Рейтинг 8/10

Google Desktop 1.0

Страсти по настольному поиску отбушевали двенадцать месяцев назад. **Грэм Моррисон** считает, что Google припозднился с предложением для Linux.

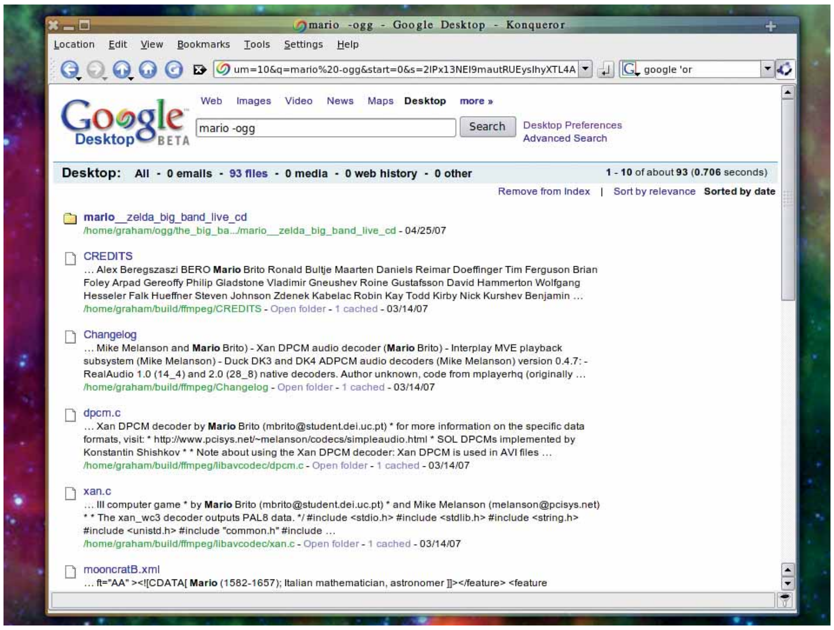
Вкратце...

» Инструмент для объединенного поиска как в Интернете, так и в пределах настольного компьютера. Может индексировать и локальные файлы, и почтовые директории на Google. См. также: *Beagle* и *Kat*.

Спустя почти три года после выпуска первоначальной версии для Microsoft Windows и на два месяца позже версии для Apple OS X, *Google Desktop* наконец-то пришел в Linux. Слово «desktop» в названии несколько двусмысленно. *Google Desktop* – только поисковый инструмент, способный заменить Kat [ныне Strigi, – прим.ред.] в KDE или *Beagle* в Gnome, но никак не рабочий стол KDE или Gnome. Хотя программа далеко не бесполезна: до сих пор нам не сильно везло с приличными поисковыми инструментами для Linux. Несмотря на потенциальную полезность, они либо занимают под свой индекс весь жесткий диск, либо отъедают солидные куски оперативной памяти, либо полностью блокируют компьютер после нескольких дней использования. Хотелось бы встретить работоспособную версию.

64 – быть?

Google предоставляет 32-битные двоичные пакеты в форматах Deb и RPM для большинства Linux-дистрибутивов. Обещана 64-битная версия, хотя мы без особого труда установили 32-битный вариант на нашу 64-битную машину. Затем остается вручную запустить программу командой *gdlinux* – на панели инструментов появляется маленький значок *Google Desktop*, и начинается важнейший процесс индексирования. Кажется, что он длится целую вечность. Индекс формируется лишь во время простоя компьютера, поэтому прошло несколько дней, прежде чем 81361 файлов на нашей машине были полностью проиндексированы, что составило около 7 часов чистого времени обработки. Подобно *Kat* или *Beagle*, поиск *Google Desktop* проникающий: просматривается и анализируется не только имя файла и его местонахождение, но и содержание, причем каждый тип файлов требует особой обработки. Поддерживается множество различных форматов с уклоном в Linux: документы *OpenOffice.org*, стандартные почтовые директории, *Thunderbird*, PDF, Ogg Vorbis, mp3 и истории браузеров. Тот, у кого есть почта на Google, может включить в индекс учетную запись Gmail, интегрировав таким образом ее с рабочим столом. Удивляет лишь полное отсутствие поддержки формата Microsoft Office.



› Приложение, в котором открывается ссылка, зависит от настроек браузера.

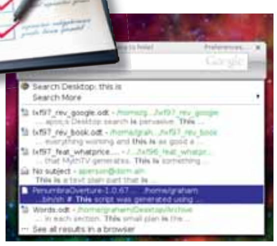
Встроенный сервер

Есть два способа начать поиск. Можно дважды нажать клавишу Ctrl, и откроется маленький поисковый инструмент, а можно вызвать новую домашнюю страницу Google из панели. Поисковый инструмент интуитивно понятнее и хорошо интегрируется в рабочий стол, подобно *Beagle* или *Spotlight* на OS X. Во время набора слов запроса результаты появляются ниже поля ввода. Нажатие Enter выводит более подробные результаты поиска в отдельном окне браузера. Выглядит все это совершенно так же, как онлайн-поиск Google, который все мы знаем и любим, включая краткий конспект по содержанию результата каждого запроса. Google утверждает, что никакие данные с персональных серверов на его собственные серверы не поступают. Типы файлов обозначаются маленькими значками, а щелчок на одном из файлов в списке найденных открывает его в назначенном по умолчанию приложении.

Все это работает безупречно, без всяких проблем с производительностью, которыми страдают «родные» Linux-приложения. Но вывод web-браузера не очень хорош с точки зрения интеграции в рабочий стол – вдобавок не работает опция ИЛИ – и возникает неприятное чувство, что это скорее циничная попытка Google навязать пользователю зависимость от своей поисковой технологии, чем действительно полезное Linux-приложение. Windows-версия обладает виджетами, чья функциональность схожа с *Dashboard/SuperKaramba*, что гораздо удобнее. Хотелось бы увидеть нечто подобное и в Linux. А пока это только очередной настольный поисковый инструмент. Пускай и работоспособный... LXF

Свойства навскидку





Интеграция

Дважды нажмите клавишу Ctrl – поисковик Google откроется и заполнит свое окно результатами поиска по вашему запросу.



Проникающий поиск

Google Search читает содержание файлов многочисленных типов и добавляет результаты чтения к своему индексу для удобства последующего поиска.

LINUX FORMAT Вердикт

Google Desktop

Разработчик: Google
Сайт: http://desktop.google.com/linux
Цена: Бесплатно (проприетарный продукт)

Функциональность	5/10
Производительность	7/10
Простота использования	7/10
Цена	7/10

» *Программа работает, она эффективна, но нет ощущения разработки специально для Linux.*

Рейтинг 6/10

Yellow Dog Linux 5.0.2

Из-за перехода Apple с PowerPC на процессоры Intel будущее Yellow Dog заволакивается туманом. **Грэм Моррисон** видит (маловероятного) спасителя в PlayStation 3 от Sony.

Вкратце...

» Свободные и коммерческие версии PowerPC-ориентированных производных Fedora Core для старых машин Apple, IBM BladCenter JS21, Cell QS20, Mercury Cell и PlayStation 3 от Sony.



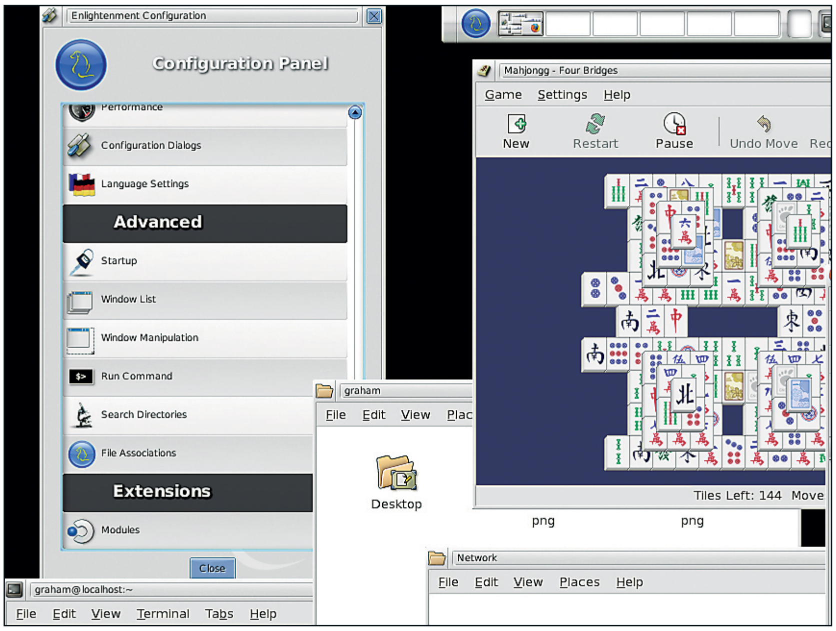
Пол считает...

Посетите сайт Terrasoft и узнайте, как мало приставок нужно собрать в YDL-кластер, чтобы достичь теоретической производительности выше одного терафлопа.

Когда Apple перешла на архитектуру Intel, сразу же встал вопрос о выживании PowerPC-ориентированных дистрибутивов вроде Yellow Dog Linux (YDL) без своей главной пользовательской базы. Но процессор PlayStation 3 (PS3) Cell (производное от PPC), вкупе с решением Sony оставить возможность установки альтернативной ОС, открывают двери для Linux. Есть версии Ubuntu и Fedora, специально собранные для PS3, но YDL была здесь с самого начала.

Другая ОС

Понадобятся обычные USB-клавиатура и мышь – мы успешно опробовали несколько разных устройств. После инсталляции Linux игровой контроллер PS3 поддерживается через USB-порт, но только как джойстик. Проблема инсталляции Linux на PS3 состоит в том, что приходится переформатировать и заново разбивать жесткий диск из главного меню приставки – инструменты дистрибутива для этого использовать невозможно, так как Sony ограничила прямой доступ и к жесткому диску, и к графической аппаратуре. Видимо, фирма опасается посягательства Linux-разработчиков на ее мировое господство в игровой сфере. Есть угроза потерять весь загруженный контент, сохраненные игры, музыку, видео и фотографии, если только предварительно не сделать резервную копию. К счастью, функция резервирования и восстановления в PS3 развита хорошо: приятно видеть, что не все потеряно, обретая прежнюю консоль после переформатирования. Сетевая учетная запись PlayStation Network сохраняется, на месте и настройка беспроводного доступа, и другие настройки, сделанные вами ранее.



› При HDMI-подключении YDL будет использовать разрешение 1280x768 для ТВ высокого разрешения, но правкой файлов конфигурации можно задействовать все 1080 пикселей.

Нужно установить загрузчик YDL с DVD (достаточно вставить диск и выбрать «Install Other OS» в системном меню PS3) и сменить ОС приставки по умолчанию, чтобы подобие загрузчика Sony изменилось на *Grub* или *LILO*. Мы обнаружили, что в случае возникновения каких-либо проблем, достаточно выключить питание консоли, затем удерживать кнопку питания в течение пяти секунд – и стандартная конфигурация Sony полностью восстановится. Инсталляция продолжительна, занимает свыше 90 минут – хотя инсталлятор и уверяет, что 25 минут для установки вполне достаточно.

Недоуправление пакетами

Набор ПО в YDL довольно скуден. Вы получаете *Firefox*, *OpenOffice.org* и *Eclipse* IDE, плюс пригоршню игр Gnome, и все это искусно вплетено в рабочий стол *Enlightenment*. Запущен и OpenSSH, а *VNC* держится наготове на случай, если вам захочется использовать ТВ. Менеджер пакетов *YUM* инсталлирован, поэтому нетрудно добавить дополнительные репозитории и загрузить все необходимые пакеты. Удобная среда с несколькими API для многопоточной Cell-разработки выглядит и работает гладко, а сравнительно небольшое потребление памяти *E17* (у PS3 только 256 МБ RAM) сохраняет отзывчивость системы. Правда, если запустить программу с чуть большими потребностями по памяти, рабочий стол начинает тормозить. *Firefox*, например, запускается 8 секунд, а после запуска еле тащится. Аппаратные ограничения Sony вынуждают Linux-дистрибутивы применять для графического вывода фрейм-буфер, не извлекая никакой выгоды из мощной аппаратуры Nvidia. Но не все так плохо, как может показаться – например, полноэкранное видео со стандартным разрешением работает без проблем. И все же графическое ускорение могло бы помочь YDL выйти на «большую дорогу» и получить более широкое признание. LXF

Свойства навскидку



Пакеты YUM
Добавьте собственные репозитории для установки новых приложений – и ваша PS3 превратится в полноценный медиаплейер.



Специально для PS3
Инсталляция ПО и поддержка сети специфичны для PS3, а значит, заставить заработать функции вроде беспроводного доступа совсем несложно.

LINUX FORMAT Вердикт

Yellow Dog Linux 5.0.2

Разработчик: Yellow Dog Linux
Сайт: www.terrasoftsolutions.com
Цена: DVD от $49.95

Функциональность	9/10
Производительность	8/10
Простота использования	7/10
Цена	8/10

» *Годится для самоделок, эмуляции и способен производить впечатление на девушек, но не заменит ПК.*

Рейтинг 8/10

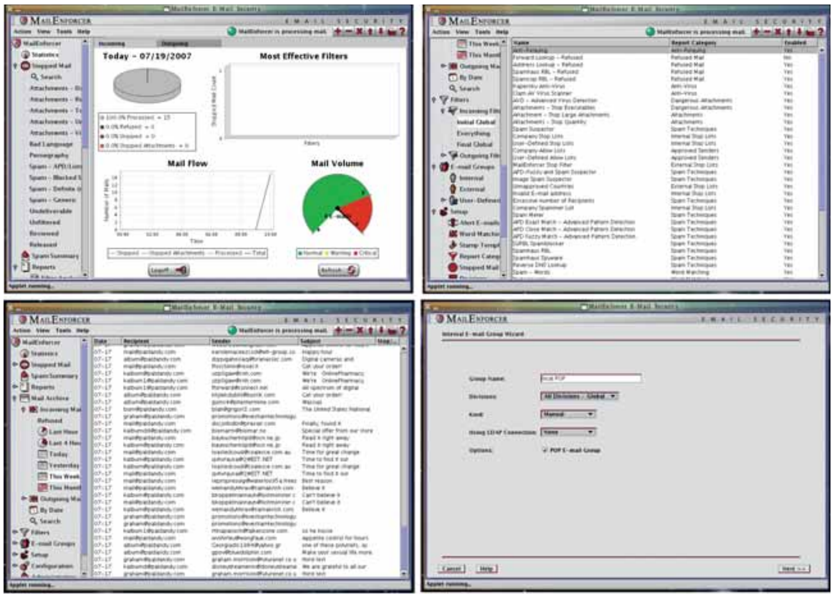
MailEnforcer 3.0

На рынке ПО для предприятий затевается нечто действительно серьезное, если **Грэм Моррисон** решился вставить в обзор выражение «проталкивать».

Вкратце...

» Почтовый прокси-сервер, фильтрующий сетевой почтовый трафик. Поставляется как на DVD, так и в составе самостоятельного сервера. Альтернативой можно назвать лишь комбинацию специальных скриптов, коммерческих антивирусов и фильтров контента наподобие *DansGuardian*.

Безопасность Linux общеизвестна; главным образом потому-то мы им и пользуемся. Но это видимое преимущество лишь сейчас начинает влиять на мир бизнеса: сразу несколько компаний готовят продукты, призванные протолкнуть Linux на широкий рынок. Устройство Yoggie Gatekeeper Pro, которое мы рассматривали в LXF95 – яркий тому пример. Это встраиваемый Linux-компьютер, установленный на USB-брелок, для фильтрации и защиты интернет-данных – при этом неважно, под какой ОС работает компьютер-хост. *MailEnforcer* имеет сходное назначение, только масштаб крупнее. Продукт поставляется предустановленным на сервер по вашему выбору либо в виде комплекта DVD/CD для инсталляции на имеющееся оборудование, с целью обеспечения безопасности электронной почты на уровне предприятия. После установки программа занимает место между почтовыми серверами и Интернетом, проверяя сообщения на соответствие строгим критериям, а заодно регистрируя все сообщения, создавая отчеты и посылая предупреждения.



› *MailEnforce* использует Java-оболочку для настройки и наблюдения за сервером; впервые СУБД Oracle снабжена экземпляром Enterprise Linux от Oracle.

Мудрость Oracle

Для установки *MailEnforcer* на наш сервер Athlon 64 X2 мы взяли DVD-версию. Фактически, устанавливается 64-битная версия Oracle Enterprise Linux, и это был первый раз, когда данный отпочкованный Oracle от Red Hat дистрибутив повстречался нам в живой природе. Литература о продукте утверждает, что глубоких познаний Linux не требуется, и это соответствует истине – по крайней мере, во время инсталляции: даются точные указания о том, что устанавливать, какие службы включить (есть только SSH и SMTP) и как сменить пароль суперпользователя. Все остальное делается через web-интерфейс, доступный после инсталляции. Однако мы не рекомендуем слепо полагаться на Linux, не обладая хотя бы минимальными навыками. Средний администратор без опыта работы в Linux может споткнуться, даже делая такие простые вещи, как смена IP-адреса сервера – особенно если инсталляция не включает X11.

Web-интерфейс – облицовка либо для встроенного Java-апплета, либо для загружаемого приложения Java Web Start (последнее работает на нашей машине немного быстрее). Успешным был и опыт с Blackdown Java – единственный метод получить 64-битную версию Web Start. Да и любая другая 32-битная инсталляция Java (1.4 и 1.5) не должна вызвать особых проблем. Во время работы внутри *MailEnforcer* крутится столько страниц, что ими можно было бы наполнить мифическую Вавилонскую библиотеку. Это пугает до тех пор, пока вы не уясните, как все-таки работает эта система. *MailEnforcer* напоминает web-прокси, а чтобы поместить программу между вашими почтовыми клиентами и серверами (как правило, *Microsoft Exchange* или Sendmail), ее нужно настроить. В малом окружении *MailEnforcer* можно использовать даже как самостоятельный POP-сервер, установив простые настройки и включив пользователей во внутреннюю группу – кстати, это неплохая возможность протестировать систему перед началом ее серьезного использования. Правильно настроенный, *MailEnforcer* должен быть прозрачен для конечного пользователя, проявляясь разве что в извещениях о блокировке сообщения, которое нужно как-то вызволять.



› **Управление почтовой системой с помощью браузера эффективно, но требует хотя бы начального уровня знаний о Linux.**

Изобилие настроек

Сказать, что *MailEnforcer* легко настраивается – значит, сказать очень мало. Под собственные нужды здесь можно перекроить буквально все. Характерный пример – фильтры сканирования почты, имеющие более сорока параметров настройки (и все включены по умолчанию). Среди них проверка коммерческим (*Kaspersky*) и открытым (*ClamAV*) антивирусами, доступ к различным онлайн-базам данных по фишингу и спаму, черные списки DNS, оптическое распознавание изображений и многие другие тесты. Мы заваливали программу сотнями писем из наших коллекций спама, а *MailEnforcer* добросовестно блокировал их или помечал как подозрительные. Страница обзора статистики, кроме своего прямого назначения, позволит вам засечь необычно высокую почтовую активность и направит на ваш электронный адрес более подробные отчеты. Вдумчиво комбинируя подразделения и группы, можно добиться осуществления любой мыслимой политики, а также переложить часть бремени администрирования, используя микроменеджмент. Продукт легко масштабируется и подходит как для малых/средних предприятий, так и для сравнительно крупных организаций. Подобной программы для Linux мы до сих пор просто не встречали. LXF

LINUX FORMAT Вердикт

MailEnforcer 3.0

Разработчик: Drawbridge Software
Сайт: www.drawbridgesoftware.com
Цена: US$995 за лицензию на 10 пользователей

Функциональность	9/10
Производительность	7/10
Простота использования	6/10
Цена	8/10

» *Прекрасный выбор для компаний всех размеров, еще не выработавших стратегию почтовой безопасности.*

Рейтинг 8/10

Paragon NTFS for Linux 6.0

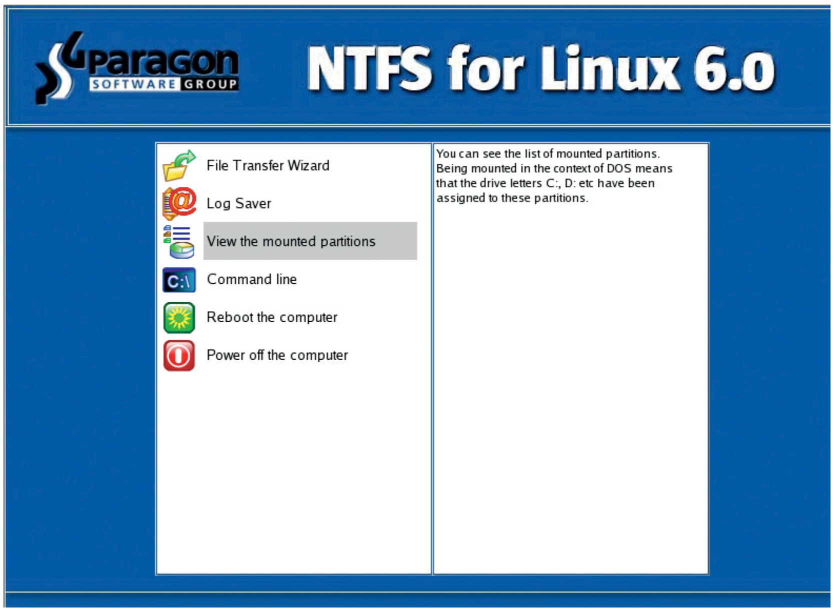
Нужна запись на жесткие диски Windows XP или Vista из-под Linux? **Нейл Ботвик** тестирует единственное потенциальное решение с поддержкой сжатия NTFS.

Вкратце...

» Коммерческая файловая система для доступа к NTFS-разделам Windows из Linux, с поддержкой чтения и записи со сжатием. См. также: *NTFS-3G*.

Пользуясь открытым ПО, мы принимаем как должное даваемую им свободу. Здесь нет тайн, и программы легко срабатываются друг с другом: ведь их внутренние механизмы показаны в исходном коде. Если нужен доступ к файловой системе Linux из Windows, достаточно лишь установить драйвер – вся информация, необходимая для его написания, открыта и доступна. Попробуете сделать наоборот – совсем другая история. В ядро Linux встраивается ограниченная поддержка файловой системы NTFS, ограниченная в том смысле, что безопасным считается только чтение с этой системы. Альтернатива – модуль *NTFS-3G FUSE*, предоставляющий доступ и на чтение, и на запись. Paragon сработал еще одну, коммерческую альтернативу, но по плечу ли ей заставить вас раскошелиться (а заодно и поступиться принципами свободного ПО)?

Существует две версии *NTFS for Linux*: персональная, о которой пойдет речь в нашей статье, устанавливает лишь драйвер файловой системы без дополнительных инструментов. Профессиональная версия оснащена инструментами для создания, изменения размеров, архивирования и других операций с NTFS-разделами из-под Linux. Обе версии могут похвалиться Live CD для доступа к NTFS-разделам, включая возможность резервного копирования данных на CD или DVD.



› **Надежность измерить трудно: копирование отдельных файлов проходило без ошибок и потерь, хотя изредка случались сбои при чтении в «глубоко вложенных» директориях.**

Производительность

Мы пробовали запустить тестовые программы *bonnie*, *bonnie++* и *iozone*, но они не работают с *NTFS For Linux*, и это показывает, что некоторые важные функции файловой системы упущены. Но вряд ли кто-нибудь заведет базу данных или почтовый сервер на NTFS-разделе, да и сверхскоростное массовое копирование/создание файлов вряд ли понадобится. Так что мы обошлись тестами попроще. Они состояли из копирования двух наборов файлов в обоих направлениях. Первый набор состоял из 4 ГБ больших видеофайлов, чтобы оценить максимальную скорость копирования. Второй набор, из 713 МБ малых файлов (49900 на два «ядерных» дерева), показывал способности обращения со сложными файловыми структурами при сравнительно малом объеме передаваемых данных. *NTFS For Linux* оказалась медленнее NTFS-3G в первом случае, при копировании файлов большого объема, и быстрее – во втором, при наличии сложной системы файлов. Полученные цифры приводятся ниже.

» Запись 4 ГБ больших файлов
NTFS for Linux 80 секунд
NTFS-3G 128 секунд

» Чтение 4 ГБ больших файлов
NTFS for Linux 207 секунд
NTFS-3G 108 секунд

» Запись 713 МБ малых файлов
NTFS for Linux 99 секунд
NTFS-3G 806 секунд

» Чтение 713 МБ малых файлов
for Linux 105 секунд
NTFS-3G 545 секунд

Как же работает программа? Прилично, и очень быстро. Файловая система устанавливается в качестве модуля ядра, поэтому перед инсталляцией необходимо иметь исходный код ядра и среду для компиляции, чего по умолчанию в большинстве дистрибутивов нет, но все необходимые компоненты присутствуют в стандартном репозитории. Установив их, инсталлировать программу нетрудно: достаточно распаковать zip-архив и запустить **sh install.sh** (не верьте документам, советующим использовать **./install.sh**, поскольку бит 'x' на нем не установлен). Инсталлятор формирует и инсталлирует модуль файловой системы, затем разыскивает NTFS-разделы и добавляет их в **/etc/fstab**. Есть возможность автоматического монтирования таких разделов при загрузке. Графического инсталлятора нет, а базовые инструкции, пожалуй, чересчур подробны, но запуск инсталляционного скрипта – это все, что вам нужно сделать.

Будучи установленной, файловая система монтируется так же, как и любая другая, но, вследствие различий с NTFS, получить такие же права доступа и владения, как в Linux, не получится. Вместо этого предоставляются параметры монтирования с указанием владельца, группы и прав доступа по умолчанию для файлов, совершенно так же, как с встроенными в ядро файловыми системами FAT.

Давление сжатия?

83-страничное руководство, доступное на сайте Paragon (но не включенное в загружаемый архив) покрывает вопросы инсталляции и различные параметры монтирования файловых систем NTFS, причем речь идет в основном об инструментах профессиональной версии. Существенное отличие от *NTFS-3G* – обращение со сжатыми файлами. Это файлы, сжатые на уровне файловой системы, а не обычные zip-архивы. Хотя читать такие файлы могут многие файловые системы, записывать в этот формат умеет лишь *NTFS for Linux* – и если у вас сжатая NTFS, ваш выбор уже сделан. LXF

LINUX FORMAT Вердикт

Paragon NTFS for Linux 6.0
Разработчик: Paragon Software Group
Сайт: www.ntfs-linux.com/home/personal
Цена: $29,95, включая 60-дневную поддержку (для РФ – 120 рублей)

Функциональность	3/10
Производительность	6/10
Простота использования	7/10
Цена	6/10

» *Приличная производительность, но не без проблем при работе со сложной структурой директорий.*

Рейтинг 6/10

Сравнение

Каждый месяц мы сравниваем тонны ПО – а вы можете отдыхать!

Клиенты IRC

Истошное перекрикивание не лишено недостатков, поэтому **Крис Хауэллз** рассматривает более уместный способ связи: с помощью клиентов Internet Relay Chat (IRC).



Paul Blachford

О тестировании

Тест проводился в системе Kubuntu 7.04 Feisty Fawn, установленной на Lenovo ThinkPad X60s с 1,5 ГБ ОЗУ (ограничено до 256 МБ) и 1,66-ГГц процессором Intel Core Duo. Программы сравнивались по следующим критериям:

» **Производительность** IRC-клиент тихонько дожидается на заднем плане, пока вам не понадобится войти в чат, поэтому он должен быть как можно более бережливым в отношении ресурсов. Мы сравнивали количество ОЗУ, используемое каждым клиентом.

» **Удобство использования и документация** Новичку IRC может показаться слегка недружелюбным. Насколько просто пользоваться клиентом: доступен ли он тому, кто не имеет базовых знаний о протоколе?

» **Функции IRC** Обладает ли клиент такими функциями, как передача файлов, администрирование каналов, приват и присоединение к нескольким сетям IRC одновременно? Можно ли расширить функциональность клиента использованием скриптов?

Наш выбор

Сразу после кончины динозавров – точнее, в 1989 году – финского программиста по имени Яркко Ойкаринен [Jarkko Oikarinen] осенило. Он создал первый сервер и клиент Internet Relay Chat, породив технологию, и по сей день популярную среди тысяч, если не миллионов, пользователей Интернета.

Рядом с современными видеоконференциями сермяжно-текстовые окна IRC, возможно, выглядят архаизмом. Но IRC распространен очень широко, благодаря необычайной простоте своего протокола и наличию IRC-клиента едва ли не в каждой ОС, снабженной стеком TCP/IP. Тысячи пользователей множества IRC-сетей плотно заняты обсуждением самых разных вопросов с единомышленниками, хоть про Linux, хоть про дела домашние.

К сожалению, научиться общению через IRC не так-то просто: команды управления сеансом по этому протоколу несколько загадочны. На наше счастье, существует немало IRC-клиентов, предлагающих упростить доступ к большинству обычных функций IRC, типа присоединения к каналу, отправки файлов и смены псевдонимов-ников.

Введение в IRC

IRC-сети, которых немало, состоят из многочисленных каналов, на которых собираются пользователи, объединенные интересом к предмету обсуждения данного канала. Как правило, каналы открыты, и присоединиться к ним может любой желающий. Присоединение к каналу позволяет общаться либо со всеми пользователями одновременно, либо вступать в частные переговоры с отдельными пользователями. Можно даже переслать файл.

Хотя графические клиенты скрывают от пользователя основы переговорного процесса, управляемого текстовыми командами, значение главных команд неплохо было бы знать. Вот несколько основных:

» **/join #channel** присоединяет к каналу под названием #channel

» **/nick ircUser** меняет ваш ник на ircUser

» **/part** покинуть текущий IRC-канал и не получать больше сообщений с него

Konversation

С легкостью поддерживает разговор.

Konversation написан с использованием KDE-библиотек, хотя и не является частью стандартного рабочего стола KDE. Заполучить его проще простого, ведь существуют пакеты для многих версий Linux – и он присутствует на нашем DVD.

После запуска *Konversation* появляется окно со списком серверов, позволяющее указать сеть IRC, к которой вам хотелось бы подключиться; каналы, присоединяться к которым следует автоматически; и ваш любимый ник.

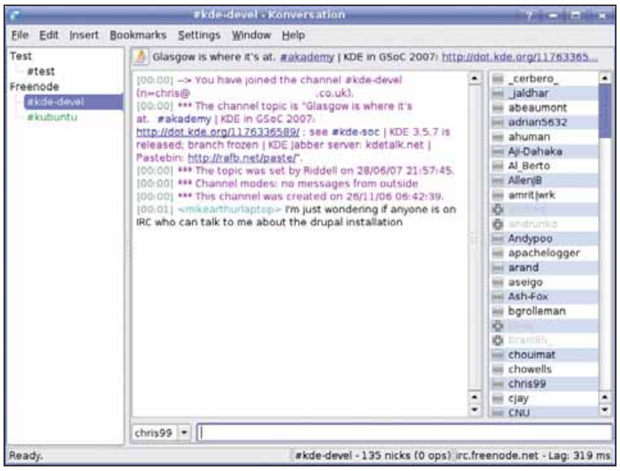
По умолчанию пользовательский интерфейс программы делится на две части, со списком сетей слева и окном чата справа. Список ников, относящихся к данному каналу, составляет часть правого окна. Различные значки используются для отображения статуса каждого ника, так что можно, например, смекнуть, кто является оператором канала.

Окно перехвата URL отображает все Интернет-адреса, упомянутые во всех каналах, к которым произошло присоединение. Есть функция Bookmark (Избранное), позволяющая «ставить закладки» на избранные IRC-каналы примерно таким же способом, как и одноименная функция браузера – в дальнейшем присоединиться к такому каналу будет значительно проще.

Для тех, кто не мыслит себя без командной строки, у *Konversation* припасена возможность встраивания эмулятора терминала Konsole. Предусмотрен и On Screen Display (OSD), где могут отображаться некоторые сообщения в то время, когда окно *Konversation* минимизировано.

Нам понравилась функция Paste Protection, которая выдает пользователю предупреждение о том, что он намеревается послать в канал больше одной текстовой строки, и позволяет отменить операцию или отредактировать содержание сообщения. Диалоговое окно настройки интуитивно понятно и разумно, поэтому барьер для нового пользователя обещает быть минимальным.

> «Paste Protection выдает предупреждение при попытке вставить больше одной строки текста.»



› ***Konversation*** **хорошо интегрируется с *KAddressBook*, связывающей ники IRC с контактами в адресной книге.**

LINUX FORMAT Вердикт

Konversation 1.0.1

Разработчик: KDE
Сайт: http://konversation.kde.org
Цена: Бесплатно под GPL

» *Солидный, удобный клиент, способный удовлетворить нужды как новичков, так и опытных пользователей.*

Рейтинг 9/10

Opera

Пакет интернет-приложений.

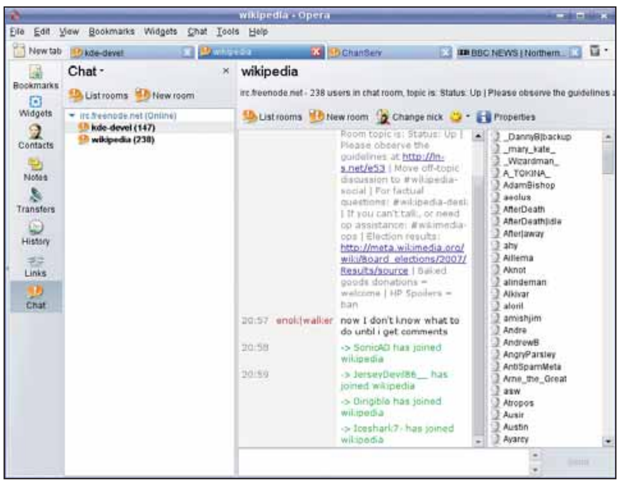
Opera IRC Client, наряду с браузером *Opera*, составляет часть пакета *Opera* Internet Suite. Все эти части интегрируются в одно цельное приложение, из которого можно читать почту, просматривать web-страницы и переговариваться в чате IRC.

Чтобы приступить к переговорам по IRC в *Opera*, достаточно внести сведения об избранной сети в окне Mail and Chat Accounts (Учетные записи почты и чатов). Настроить существенные детали, такие как ник и название сети, к которой следует присоединиться, помогает мастер. По окончании настройки с сервера загружается список сетей IRC, который отображается в окне браузера, хотя название канала можно указать и непосредственно, если вы его уже знаете.

Набравшись опыта в IRC и став оператором канала, вы обнаружите функции, позволяющие контролировать каналы и деятельность пользователей на них. Щелчок правой кнопкой мыши на одном из ников в списке пользователей вызывает контекстное меню, одно из подменю которого, Control (Контроль), содержит команды, позволяющие изгнать пользователя канала – временно (kick) или навсегда (ban).

Подключиться к новой чат-комнате можно командой **/join** или воспользовавшись немного невнятно названной кнопкой панели инструментов New Room. Нам понравилась идея работы в полноэкранном режиме, когда весь экран превращается в один гигантский IRC-сеанс, хотя при этом – досадно! – отсутствуют средства наблюдения за активностью на других каналах.

Opera работает сравнительно неплохо, хотя мы заметили несколько мелких неувязок вроде появления кнопок Source и Validate (Источник и Проверить) в меню правой кнопки мыши – эти кнопки актуальны лишь при просмотре web-страниц в браузере. А еще раздражает не замеченное нами ни в одном другом IRC-клиенте стремление *Opera* скрыть префикс # в названии канала.



› **Каналы распределяются по вкладкам: на каждой вкладке есть браузер сети/канала (слева), область канала (посередине) и список ников (справа).**

> «Полноэкранный режим превращает весь экран в гигантский IRC-сеанс».

LINUX FORMAT Вердикт

Opera 9.21

Разработчик: Opera Software, Осло
Сайт: http://www.opera.com
Цена: Бесплатно (проприетарный продукт)

» *Ничем особо не выделяется, но определенно стоит внимания пользователей браузера Opera.*

Рейтинг 6/10

Irssi

Название не звучит? Ну, это не главное!

Хотя *Irssi* – консольная программа, и назвать ее интуитивно-понятной язык не поворачивается, приложение достаточно мощное и гибкое в настройке. За ним стоит мощная команда разработчиков и пользователей, благодаря чему на сайте и в wiki программы накоплен солидный объем документации.

Настроить *Irssi* можно двумя способами. Первый – редактирование файла **~/.irssi/config** в обычным текстовом редакторе. Второй – ввод команд в программу. Работающую конфигурацию затем можно сохранить в файле настройки.

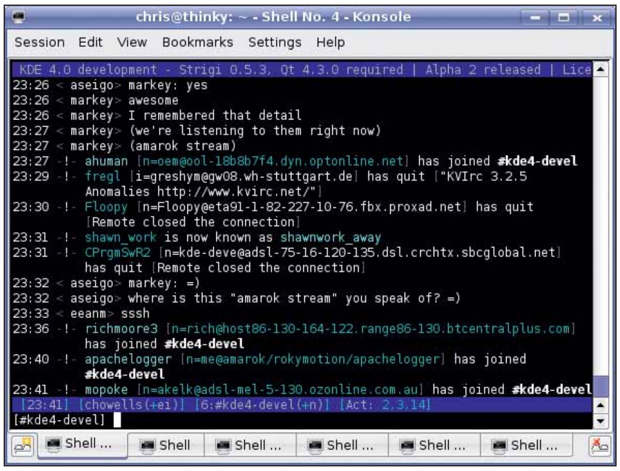
Irssi обладает поразительно широкими возможностями настройки поведения, как с использованием визуальных тем, так и с помощью скриптов. Скрипты пишутся на Perl. Немало разнообразных тем и готовых скриптов доступно на сайте программы. Одно из скрытых достоинств *Irssi* – модуль *irssi-proxy*, который, согласно своему названию, действует как IRC прокси-сервер. Когда *Irssi* работает на сервере, к нему можно подключиться с другого компьютера – например, вашего настольного PC – и если вы выключаете компьютер на ночь, оставляя сервер в работе, нет нужды отсоединяться от IRC.

Если вам случалось по ошибке садануть большую вставку текста в чужой IRC-канал, вам будет приятно узнать, что при отправке более чем пары строк *Irssi* сперва обязательно потребует подтверждения такого действия.

Приятное впечатление произвела способность *Irssi* к самообновлению. По команде /upgrade программа автоматически загружает новые бинарники, на удивление даже без выхода и отключения от IRC-серверов.

Наблюдение за активностью IRC-каналов реализуется с помощью списка названий активных окон в правом нижнем углу строки состояния. Если кто-нибудь в канале или привате обращается непосредственно к вам, то номер окна окрашивается в пурпурный цвет.

> «Одно из скрытых достоинств Irssi — модуль irssi-proxy.»



› Объедините *Irssi* с GNU screen (www.gnu.org/software/screen) и SSH – и сможете отовсюду подключаться к одному и тому же IRC-сеансу.

LINUX FORMAT Вердикт

Irssi 0.8.11

Разработчик: Тимо Сираинен
Сайт: www.irssi.org
Цена: Бесплатно под GPL

» *Irssi вряд ли подойдет новичку, но для опытного пользователя трудно отыскать более подходящий IRC-клиент.*

Рейтинг 8/10

Pidgin

Много протоколов в единственном интерфейсе.

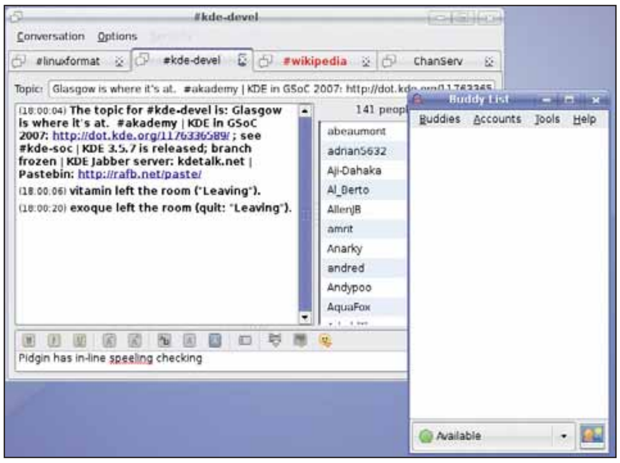
Вы были когда-нибудь в буфете «съешь, сколько сможешь» и, поняв название буквально, проклинали себя в течение нескольких следующих часов? Таким мы нашли *Pidgin*. Известный прежде как Gaim, *Pidgin* – это клиент сети мгновенных сообщений, поддерживающий множество разных протоколов, включая IRC.

Всеядность *Pidgin* в поддержке самых разных протоколов вынуждает кое-чем пожертвовать ради поддержки IRC. Ведь основная задача программы – обмен персональными мгновенными сообщениями, а это большая разница.

Добавив IRC-сервер к настройке учетной записи, мы угодили в ловушку: не обнаружилось ни окна для ввода сообщений, ни возможности присоединиться к каналу. В конце концов мы догадались, что название канала необходимо указать в меню Buddies > Join A Chat (Друзья > Присоединиться к чату). После этого появилось новое окно, используемое только для IRC-функций *Pidgin*. Каждый канал занимает отдельную вкладку с выводом текста слева и списком ников – справа.

Интерфейс *Pidgin* примитивен, а набор функций ограничен. Мы не нашли способа сменить свой ник, кроме обращения к команде /nick, а если не возвращаться в главное меню программы, то нет способа присоединиться к новому каналу, кроме команды **/join**. Можно добавлять каналы к списку Buddies (Друзья), хотя немало разочаровывает, что эта функция не автоматизирована. К сожалению, команда /server тоже не реализована. А еще *Pidgin* не позволил нам присоединиться к Freenode с нашим обычным ником, ни с того ни с сего потребовав заменить последний знак единицей.

> «Поддержка большого числа протоколов требует жертв.»



› Репутацию *Pidgin* отчасти спасает встроенная проверка орфографии, подсвечивающая неверно набранное слово в поле ввода текста.

LINUX FORMAT Вердикт

Pidgin 2.0.2

Разработчик: Pidgin (ранее Gaim)
Сайт: http://pidgin.im
Цена: Бесплатно под GPL

» *IRC-клиент очень неудобен в пользовании и не особенно-то сочетается с остальной частью Pidgin.*

Рейтинг 5/10

KVIrc

Клиент с богатыми возможностями настройки.

Еще один клиент, написанный с использованием библиотек KDE и Qt. Буква "V" в названии *KVirc*, видимо, означает "visual", но, хотя выглядит *KVirc* весьма неплохо, не совсем ясно, почему визуальность считается таким уж важным качеством.

По умолчанию *KVirc* оснащается интерфейсом MDI (Multiple Document Interface), а не современным вкладочным, в стиле *Firefox*. Поэтому каждый канал размещается в отдельном окне внутри главного окна программы. Окно канала можно перемещать, минимизировать и максимизировать; можно даже вывести его из главного окна и разместить в любом месте рабочего стола.

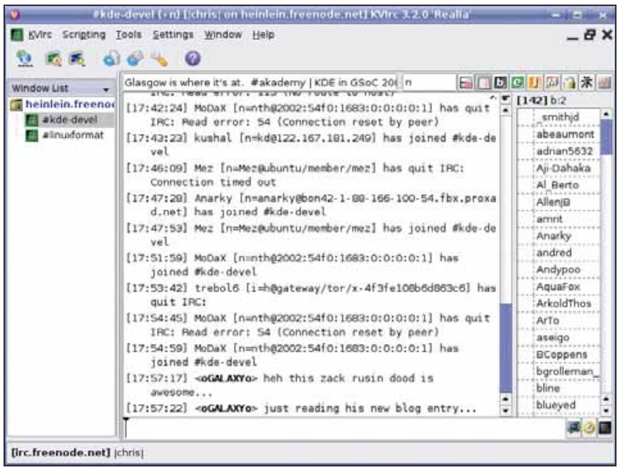
Похоже, что *KVirc* рассчитан на сравнительно опытных IRC-пользователей, судя по таким функциям, как редактор действий и синонимов-алиасов. Редактор алиасов позволяет создавать клавиатурные комбинации, с помощью которых можно выполнять весьма сложные действия благодаря встроенному скриптовому движку.

Хотя в целом документация *KVirc* довольно подробна, эти функции в ней освещены скудно; очень жаль. Зато документация по скриптовому движку составлена превосходно.

Пользовательский интерфейс *KVirc* интуитивно-понятен и гибок в настройке, включая поддержку разнообразных тем, многие из которых доступны для загрузки с сайта *KVirc*. Впечатляют и способности программы к шифрованию. Поддерживаются многие протоколы шифрования, включая *Mircryption*, применяемый также в *XChat* и популярном IRC-клиенте для Windows mIRC. Шифрование применяется в каналах, в привате и прямых чатах DCC (Direct Client-to-Client).

Высоко развита поддержка скриптов: в *KVirc* есть даже собственный скриптовый движок под названием KVS (*KVirc* Scripting language), обладающий элементами различных языков программирования, включая C++ и Perl. Преимущество языка состоит в том, что он создан специально для работы в IRC. Но при желании можно составлять скрипты на чистом Perl.

> «Развита поддержка скриптов: для них даже есть свой движок.»



› Справочная система *KVirc* делит окно пополам. Продолжая работать с программой в левой части, можно просматривать документацию в

LINUX FORMAT Вердикт

KVIrc 3.2.0

Разработчик: Шимон Стефанек
Сайт: www.kvirc.net
Цена: Бесплатно под GPL

» *Местами невразумителен, но встроенная поддержка скриптов весьма удобна для автоматизации.*

Рейтинг 6/10

XChat

Этот клиент на GTK+ – хороший работяга.

Сверхпопулярный клиент, отчасти благодаря долгожительству (разрабатывается еще с 1990-х годов), отчасти из-за кроссплатформенной архитектуры.

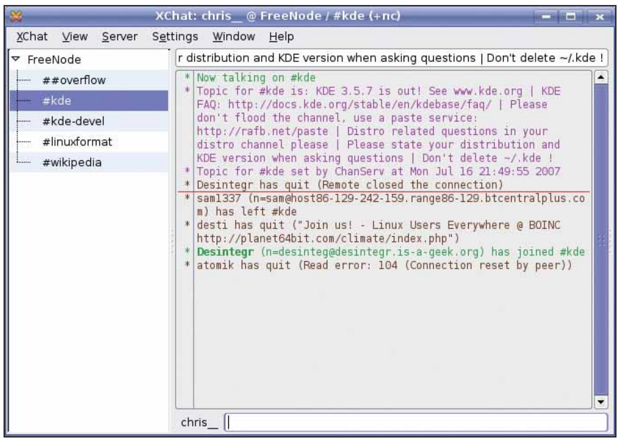
После первоначального запуска *XChat* вам необходимо указать сеть, к которой нужно подключиться, хотя эта стадия не особенно приветлива, а набор функций прямо-таки устрашает.

Программа предоставляет широкие возможности настройки интерфейса. По умолчанию в левой части окна представлены серверы и каналы в виде дерева, в середине – главное окно чата, а справа остается список ников. Можно отказаться от «древесного» вида ради вкладочного варианта, с корешками вкладок для каждого канала в низу экрана. Приятная функция – горизонтальная черта, которая подводится под текущим чатом при переключении на другой канал: по возвращении обратно сразу видно, сколько всего произошло за время вашего отсутствия на канале.

Если кто-нибудь на канале упоминает ваш ник (или слова из настраиваемого списка для слежения), *XChat* старательно пытается привлечь ваше внимание, включая мигание кнопки на панели задач KDE. Заодно в системном лотке появляется значок *XChat*, хотя не совсем ясно, зачем: ведь единственный параметр, который можно при этом включить – это "Quit" (Покинуть программу). Подобным образом эти функции работают и в Gnome.

Клиент располагает обширным набором клавиатурных комбинаций, что делает возможным управление с помощью клавиатуры, почти без участия мыши. Наиболее удобна комбинация Alt+N (где N – цифра от 0 до 9) для переключения между каналами. Есть возможность назначить собственные комбинации, если умолчания не устраивают.

> «Набор клавиатурных комбинаций позволяет обойтись без мыши.»



› Скрипты – важная функция *Xchat*, их можно писать и на Perl, и на Python; много готовых скриптов имеется на сайте программы.

LINUX FORMAT Вердикт

XChat 2.8.4

Разработчик: Петер Железны
Сайт: www.xchat.org
Цена: Бесплатно под GPL

» *Зрелый солидный клиент, умело балансирующий между простотой использования и мощью.*

Рейтинг 9/10

Chatzilla

Лучший выбор для «сталкера»!

Этот клиент можно получить либо в составе браузера *Seamonkey*, либо как расширение к *Firefox*. Для данного **Сравнения** мы взяли модуль *Firefox*, посчитав, что именно так его могло бы использовать большинство читателей.

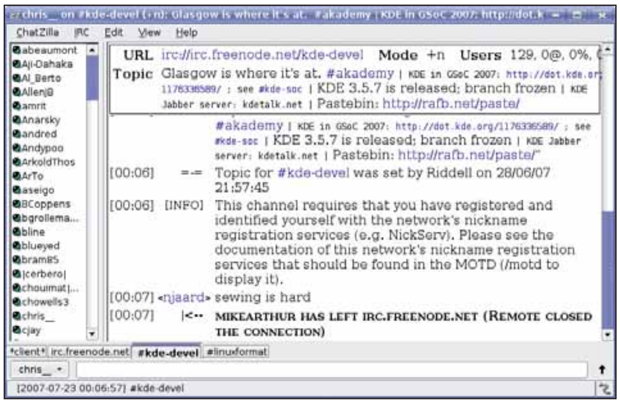
Если *Firefox* уже установлен, то инсталляция предельно проста – всего-навсего войдите на сайт *Chatzilla* и нажмите установочную ссылку (или возьмите файл с DVD этого номера). После загрузки и инсталляции нужно перезагрузить *Firefox*, после чего *Chatzilla* появится среди пунктов меню Tools (Сервис). Другой способ запуска – ввод в адресной строке *Firefox* «Интернет-адреса» в таком формате: **irc://server/#channel**.

Некоторые раздражающие черты *Chatzilla* застали нас врасплох. Несколько сетей прописаны в программе по умолчанию, и одного щелчка на их названии достаточно для подключения к IRC-серверу. Но совсем не так просто добавить к списку новую сеть. В диалоговом окне Preferences (Параметры) есть кнопка Add (Добавить), после нажатия на которую можно добавить Network (Сеть) и Target (Канал), но, к несчастью, после добавления сети она не появляется в списке преднастроенных сетей, как можно было бы предположить.

Печально, что документация не может оказать сколько-нибудь существенной помощи. Есть, правда, объемистый список FAQ, но ничего похожего на пошаговые инструкции о том, как осуществить какое-либо действие.

Что касается технологии, то *Chatzilla* написан на JavaScript, а его пользовательский интерфейс создан с помощью каскадных таблиц стилей (CSS) [корректнее было бы сказать, XUL и CSS, – *прим. ред.*]. Альтернативные цветовые схемы, известные как Motifs, доступны на сайте программы. Нарастить функциональность *Chatzilla* можно, создав собственные скрипты JavaScript. Расширяемость программы такова, что можно добавлять даже собственные пункты меню.

> «Написан на JavaScript, а интерфейс пользователя применяет CSS».



› Не особенно богат набор функций, зато удачно именованный “Stalk list” позволяет составить список ников или текстовых строк, об упоминании которых программа должна немедленно уведомлять вас.

LINUX FORMAT Вердикт

Chatzilla 0.9.78.1

Разработчик: Chatzilla
Сайт: http://chatzilla.hacksrus.com/
Цена: Бесплатно под GPL/MPL/LGPL

» *Гибок в настройке, но хромает интеграция с браузером, по сравнению с Opera. Удобство пользования тоже не на высоте.*

Рейтинг 6/10

Kopete

AIM, ICQ, MSN, Yahoo, Jabber, IRC, Gadu-Gadu, Novell GroupWise и все, все, все…

Оформив свою учетную запись в *Kopete*, вы оказываетесь почти в такой же ситуации, как и в *Pidgin*: необходимо вернуться назад, в главное меню, и выбрать File > Add Contact (Файл > Добавить контакт). Затем нужно будет указать ник или название канала. Правда, оформлено все это сравнительно приятно: ник или название канала сопровождают вас во время переговоров по ICQ, AIM или MSN Messenger (да и по всем остальным IM-протоколам, которые поддерживает *Kopete*).

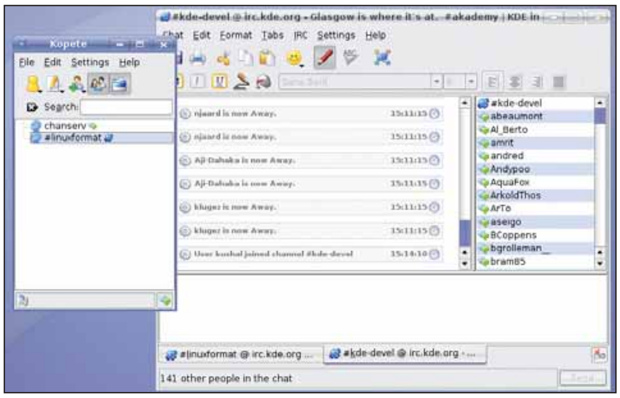
Хотя опытный IRC-пользователь вряд ли оценит такую способность, но *Kopete* гораздо лучше *Pidgin* осуществляет интеграцию IRC-клиента в унифицированный клиент сети мгновенных сообщений.

Начать приватные переговоры с кем-нибудь из своих собеседников можно, дважды щелкнув по его нику в главном окне программы; присоединиться к какому-либо каналу можно подобным же образом. После присоединения к каналу в окне чата возникает отдельная вкладка, но окно сообщений делится на вывод канала слева и список ников справа.

Щелчок правой кнопкой мыши в списке ников выдает контекстное меню, которое оператор IRC-канала может использовать для управления.

Программа хорошо интегрируется с остальными компонентами KDE, такими как адресная книга, поэтому любой контакт из адресной книги можно связать с его IRC-ником. Сделав это, можно отправлять сообщения на электронный адрес абонента прямо из *Kopete* – очень удобно для связи, например, с коллегами при работе над распределенным проектом.

> «Неплохо интегрирует IRC в унифицированный клиент сообщений».



› Увы, как и в *Pidgin*, команда /server не реализована, и для подключения к другой сети необходимо создавать новую учетную запись. Другие IRC-команды – например, /join для присоединения к новому каналу – действуют.

LINUX FORMAT Вердикт

Kopete 0.12.1

Разработчик: KDE
Сайт: http://kopete.kde.org
Цена: Бесплатно под GPL

» *Хорошо интегрирует IRC с другими IM-протоколами в надежном и удобном клиенте.*

Рейтинг 7/10

IRC-клиенты

Вердикт

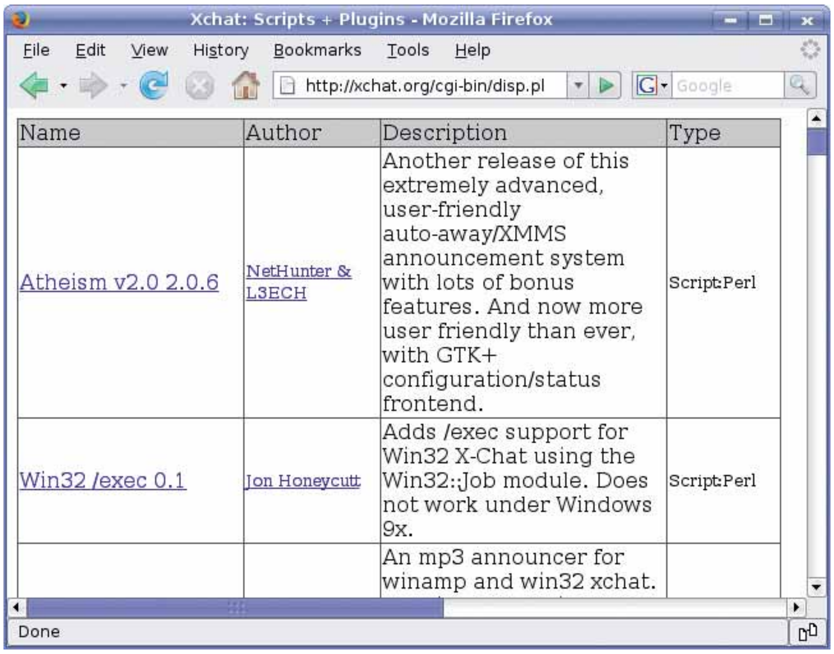
Xchat 9/10

Фактически, IRC не особо изменился с 1989 года, оставаясь до сих пор чрезвычайно простым протоколом – простым до такой степени, что для участия в чатах достаточно telnet-клиента (хотя это и не рекомендуется: ведь вам нужно будет отвечать на ping-запросы сервера, а если вы не будете делать это достаточно шустро – вас просто отключат!).

По счастью, все клиенты на данном тесте гораздо хитрее telnet, хотя удобство их использования существенно различается: некоторые требуют знания IRC-команд, тогда как в других почти все делается из графического интерфейса.

Из клиентов, проходивших наш тест, благоприятное впечатление на нас произвели три. Это *XChat*, *Konversation* и *Irssi*.

XChat – клиент, успешно сочетающий многие из наиболее важных аспектов IRC в единственной программе, достаточно простой для новичков и обладающей богатыми скриптовыми возможностями. Клиент базируется на *GTK+* и почти одинаково работает в Gnome и KDE. И все же пользователям Gnome полезно было бы взглянуть на альтернативную версию *XChat* под названием *XChat-gnome* (http://xchat-gnome.navi.cx), лучше интегрированную с Gnome. Будучи «клиентом для всех», *XChat* заслуженно получил высшую награду нашего теста.



› На сайте *XChat* доступны к загрузке сотни готовых скриптов. А если нужного не найдется – почему бы не написать что-нибудь самому на Perl или Python?

Соискатели

Konversation – превосходный клиент для KDE, сочетающий выдающееся удобство использования с замечательной документацией. Но хотя эта программа может использовать скрипты с помощью dcop (программа *kdcop* выведет все доступные виды интерфейсов), ее скриптовые способности далеки от аналогичных возможностей *XChat*. Поэтому он идет голова к голове с *Irssi*.

Irssi – единственный консольный клиент в нашем обозрении. Даже если вы не совсем в ладах с оболочкой, мы настоятельно рекомендуем поэкспериментировать с ним и почитать man, прежде чем изгонять эту программу с компьютера за то, что она недостаточно понятна интуитивно. Вы наверняка заметите, что небольшое количество потраченного таким образом времени экономит многие часы в дальнейшем. *Irssi* вполне заслуженно разделила в нашем сравнении второе место с *Konversation*.

Chatzilla – единственный клиент нашего тестирования, который нам не хотелось бы рекомендовать кому бы то ни было из-за его раздражающего поведения, неудобства в использовании и чрезвычайно скудной документации. LXF

Вопрос читателю

Вы фанат IRC или предпочитаете другие средства, например, ICQ, AIM или MSN Messenger? Сообщите нам свое мнение и поделитесь соображениями на letters@linuxformat.ru. Кстати, почему бы не присоединиться к #linuxformat на irc.freenode.net?

Таблица функций

Приложение	Версия	Лицензия	Более одного сервера	Скрипты	Темы	Ведение журнала	Проверка орфографии	Поддержка разных протоколов	Потребление памяти
KVIrc	3.2.0	GPL	✓	✓	✓	✓	☒	☒	8.0МБ
Konversation	1.0.1	GPL	✓	Частично	✓	✓	✓	☒	6. МБ
Kopete	0.12.	GPL	✓	☒	✓	✓	✓	✓	.3МБ
XChat	2.8.	GPL	✓	✓	✓	✓	✓	☒	5. МБ
Pidgin	2.0.2	GPL	✓	☒	✓	✓	✓	✓	12. МБ
Chatzilla	0.9. 8.1	GPL/MPL/LGPL	✓	✓	✓	✓	☒	☒	39.8МБ
Opera	9.21	Проприетарная	✓	☒	☒	☒	✓	☒	22.9МБ
Irssi	0.8.11	GPL	✓	✓	✓	✓	Скрипт	☒	2.5МБ

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux



ЛАДИСЛАВ БОДНАР
основатель, редактор, начальник и сотрудник DistroWatch.com.

Пингвиний угол

Представьте на миг, что вы утром проснулись и обнаружили себя в Бутане, горном королевстве у могучих Гималаев. Вас осеняет, что компьютеры важны для успеха в жизни. Однако чтобы изучать компьютеры, сначала необходимо овладеть иностранным языком. Такова печальная экономическая реальность: Microsoft не делает Windows для Бутана просто потому, что этот продукт столь низкого спроса не окупит вложенные в перевод и локализацию средства.

Вечный родник надежды

К счастью, для Linux и другого ПО с открытым кодом, укоренившимся в Интернете, ситуация не столь безнадежна, как раньше. Правительства различных небольших и малочисленных стран подхватили инициативу создания локализованных дистрибутивов Linux. В Бутане министром Информации и Коммуникаций недавно был запущен основанный на Debian Dzongkha Linux (джонгха – официальный язык Бутана). В соседнем Непале подобные усилия вылились в разработку NepaLinux, также основанного на Debian, полностью локализованного на непальском.

Там, где этой работой не управляют правительства, инициаторами часто выступают сообщества разработчиков Linux. Прекрасный пример – Насао, вьетнамский дистрибутив на основе Puppy Linux: он не только полностью переведен на вьетнамский, но также мал, быстр и комфортно чувствует себя даже на скромном оборудовании.

Мы часто называем Linux альтернативной операционной системой. Для многих людей, однако, Linux далеко не альтернатива – это единственная доступная для них ОС.

ladislav.bodnar@futurenet.co.uk

Скоро выйдут...

С ростом популярности Linux дистрибутивы наперебой предлагают новые функции. Что будет на нашем столе в 2008-м?

Хотя каждый лидирующий проект вносит инновации, заодно заимствуя идеи у соперников, все дистрибутивы движутся к одной общей цели – действительно удобному настольному решению.

OpenSUSE 10.3

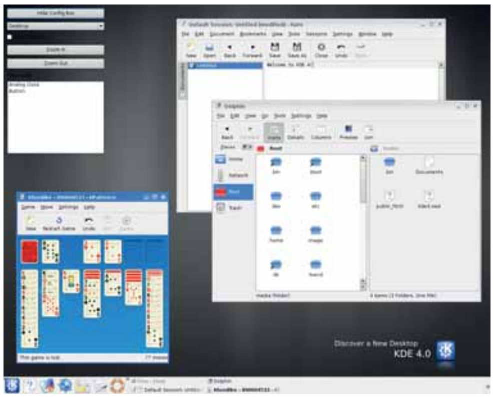
Из всех основных дистрибутивов, openSUSE ближе всего на пути к новому релизу. Версия 10.3, запланированная на 4 октября 2007, впервые включит устанавливаемые LiveCD и единственный установочный CD (с Gnome или KDE). Разработчики приступили к интеграции в дистрибутив предстоящего KDE 4, но поскольку новая версия KDE выйдет позже, чем OpenSUSE 10.3, по умолчанию останется KDE 3.5. Среди других интересных улучшений и новых функций – интегрированные внешние репозитории пакетов в *YaST*, новый добавочный CD с образовательным ПО, помощник миграции и чрезвычайно ускоренное время загрузки.

Ubuntu 7.10

Обычный 6-месячный цикл продолжается в виде 7.10, с кодовым именем "Gutsy Gibbon" [Бесстрашный Гиббон]; планируемая дата выхода – 18 октября. И вновь проект попытается включить (причем по умолчанию) 3D-рабочий стол, на этот раз благодаря недавнему слиянию проектов *Compiz Fusion*. Многое зависит от стабильности и удобства использования *Compiz Fusion*: предыдущая итерация Ubuntu в последний момент отбросила эту функцию, решив, что *Beryl* (ныне часть *Compiz Fusion*) содержит много ошибок и недостаточно устойчив. Ubuntu 7.10 также будет впервые поставляться с редакцией Mobile And Embedded, нацеленной на небольшие, мобильные и встраиваемые устройства.

Mandriva 2008

Ожидается в октябре 2007. В отличие от OpenSUSE и Ubuntu, компания помалкивает о новых функциях [краткое и полное описание версии 2008 появилось в Интернете в самом конце августа 2007, – *прим. пер.*]: на момент написания дневник разработки весьма поверхностен, а Wiki содержит лишь предполагаемый список, а не окончательные решения. Тем не менее, Cooker (дерево разработки Mandriva) регулярно обновляется, а последний пресс-релиз намекает на возможную интеграцию *NEPOMUK*, технологии социального семантического рабочего стола, в окружение KDE. Кроме стандартного обновления ядра и пакетов, говорится также о лучшей интеграции в Mandriva 2008 легковесного рабочего стола *Xfce*.



› Все основные дистрибутивы сулят поставку давно предвкушаемого KDE 4, но будет ли он стабилен?

Fedora 8

В данный момент назначенный на 7 ноября, выход Fedora будет последним релизом основных дистрибутивов в этом году. Он будет иметь важное преимущество по времени для интеграции KDE 4 и исправления любых потенциальных ошибок. Что касается пользователей Gnome, одной из неожиданных новых функций является замена традиционной панели Gnome на *Online Desktop* (часть проекта *Mugshot*), разработку, спонсируемую Red Hat и интегрирующую различные сетевые сервисы, типа публикации фотографий и web-почты, в улучшенную панель. Кое-что еще улучшено и отполировано, но радикальных новшеств нет; похоже, Fedora 8 – куда менее амбициозный релиз, чем Fedora 7.

Все эти дистрибутивы будут поставляться с последними версиями основных компонентов и пакетов; вероятно, с ядром 2.6.22 и множеством новых драйверов для беспроводных сетевых карт; а графическая система непременно получит в подкрепление *X.Org 7.3*. Хотя KDE 4 все еще вызывает беспокойство по поводу времени выпуска и качества окончательного кода, Gnome это не касается: все четыре дистрибутива предложат Gnome 2.20 (обещанный на 19 сентября). *OpenOffice.org* должен в сентябре обновиться до 2.3, но *Firefox 3.0*, ожидаемый в октябре, вряд ли успеет попасть в любой из четверки главных дистрибутивов.

Только серфистам

Webconverger Дистрибутив Linux, нацеленный только на web-серфинг.

Вы встречали их в интернет-кафе: бросаете в прорезь монетку, и открывается web-браузер, предоставляя вам неограниченный доступ к богатствам web. Но попытайтесь сохранить файл или запустить исполняемую программу, и обнаружите темную сторону этих так называемых интернет-киосков: они разработаны только для web-серфинга, и любые другие действия, ожидаемые от компьютера, запрещены. Хотя, при ограниченной функциональности, у них также есть важное преимущество: они надежны и практически не требуют сопровождения.

Компания, разрабатывающая альтернативу с открытым кодом для более распространенных интернет-киосков, основанных на Windows – базирующаяся в Великобритании Webconverger. Ее продукт, носящий то же имя, работает как мгновенно используемый дистрибутив Linux, запускаемый с CD. При загрузке он автоматически настраивает интернет-соединение, затем открывает *Firefox*. Браузер не имеет панели инструментов и не может быть закрыт или свернут мышкой, но предусмотрена комбинация клавиш для закрытия браузера, на случай его некорректного поведения. Все обычные для рабочего стола Linux комбинации клавиш отключены. По части модулей расширения, он поставляется с преднастроенными *Flash Player 9* и *MozPlugger* для проигрывания файлов *QuickTime*.



› **Webconverger, дистрибутив, разработанный для интернет-киосков, основан на Debian Live.**

Свободный демо-CD доступен для загрузки на сайте компании. Для внедрения в интернет-кафе или другие коммерческие среды, Webconverger предлагает настраиваемое решение с дополнительными функциями: домашней страницей, возможностью печати web-страниц, поддержкой ввода интернациональных символов и фильтрацией контента.

http://webconverger.com

Вопросы темпа

Yoper Linux 3.0 Согласно его разработчикам, это быстрейший дистрибутив Linux на свете.

Появившийся в Новой Зеландии, он не только скомпилирован и оптимизирован для архитектур современных процессоров: его ядро улучшено заплатками, которые большинство остальных дистрибутивов не приняли, опасаясь нестабильности. К тому же многие ключевые приложения и библиотеки предварительно загружаются в память для снижения времени запуска. Все это и другие маленькие хитрости дают ощущение, что все на рабочем столе Yoper происходит быстрее.

Скорость – не единственный фактор: часть самых свежих пакетов Yoper поставляется через RPM, но его менеджер пакетов основан на *Smart*, системе управления ПО с автоматическим разрешением зависимостей и графической оболочкой. Просматривая установленное и доступное ПО, вы увидите значительный массив приложений, библиотек и драйверов (по последним подсчетам, свыше 1500), и проприетарных, и свободных, включая графические драйвера ATI и Nvidia, модули *Adobe Flash*, Java, мультимедийные проигрыватели и кодеки, пакеты 3D-рабочего стола (*Beryl* и *Compiz*) и стандартный набор универсальных программ с открытым кодом для рабочих столов.



› **Оригинальное оформление, тема рабочего стола и обои создают интересное визуальное ощущение в Yoper 3.0.**

Начав как коммерческий дистрибутив, Yoper позднее превратился в сообщество разработчиков-энтузиастов, ведомых Тобиасом Гершнером [Tobias Gerschner], ради продолжения работы. Первым стабильным релизом проекта за два года стал Yoper 3.0, вышедший в июне 2007 после нескольких месяцев бета-тестирования. Как и следует ожидать после столь долгой задержки, новая версия весьма превосходит предыдущие релизы проекта; теперь он доступен как установочный и LiveCD и обзавелся улучшенной функцией автоматического обнаружения устройств.

www.yoper.com LXF

Экзотические дистрибутивы

Linux добрался до всех уголков Земли. Вот несколько наиболее экзотических дистрибутивов, поддерживаемых правительствами, небольшими компаниями и сообществами энтузиастов.

Ankur Bangla	Основан на Mandriva с поддержкой бенгальского, на котором говорят в Бангладеш и районах Индии. www.bengalinux.org
BOSS GNU/Linux	Основан на Debian; его амбициозная цель – полная поддержка всех официальных языков Индии. http://bosslinux.in
Hacao Linux	Вьетнам, основан на Puppy Linux. Быстр, мал и полностью переведен на вьетнамский. www.hacao.com
Karamad Linux	Иран, основан на Slackware. Разработан для людей, интересующихся языком и культурой древней Персии. www.karamad.com
NepaLinux	Основан на Debian, полностью локализован на непальский. www.nepalinux.org
Soyombo Linux	Основан на Debian, Монголия; правда, на момент написания проект уже более двух лет дремлет. http://openmn.sf.net
Swecha GNU/Linux	Основан на Debian, разработан в Индии; локализован на телугу, государственном языке штата Андхра Прадеш. www.swecha.org
Utkarsh	Индия, основан на Debian, поддерживает говорящих на гуджарати. www.utkarsh.org
Wazobia Linux	Нигерия, коммерческий, основан на Red Hat. Поддерживает английский, хауса, йоруба и игбо. www.wazobialinux.com

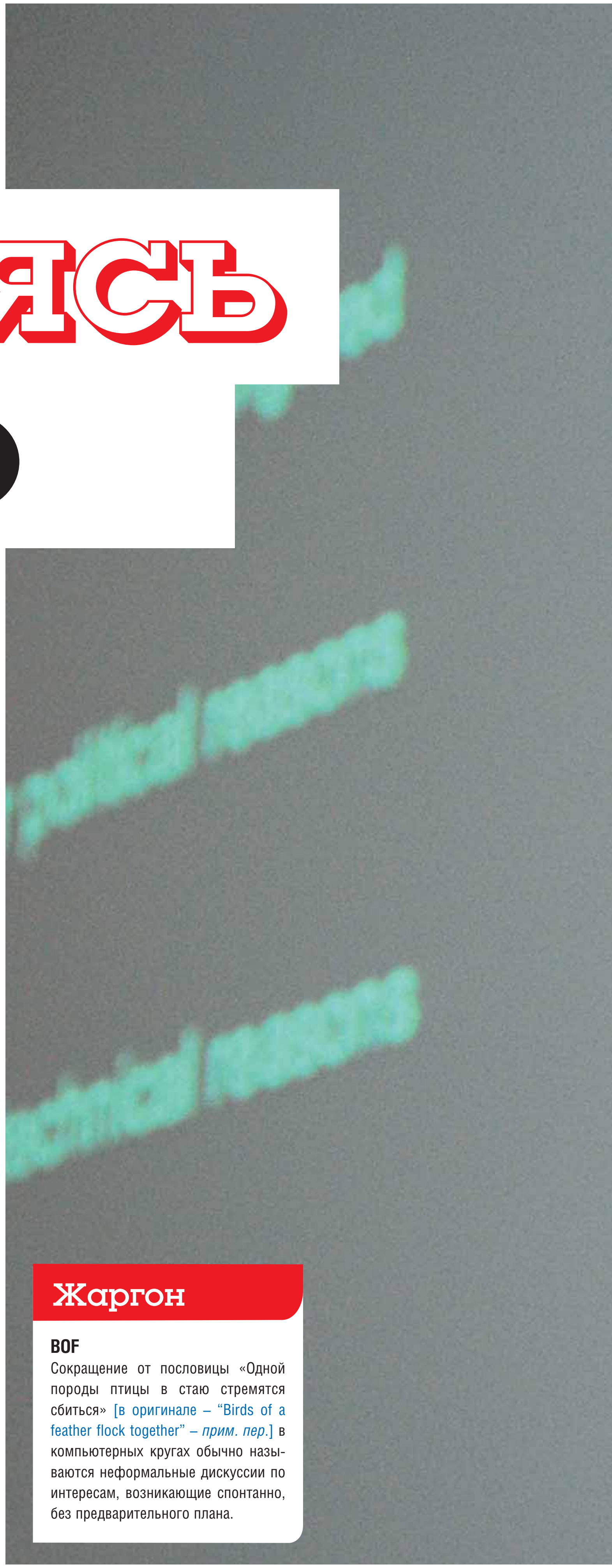
Хит-парад

10 самых посещаемых страниц на DistroWatch.com с 24 июля 2007 по 1 августа (среднее число визитов в день)

Дистрибутив	Число визитов	
1 PCLinuxOS	1,997	↑
2 Sabayon	1,937	↓
3 Ubuntu	1,935	↓
4 OpenSUSE	1,191	↑
5 Mint	1,145	↑
6 DesktopBSD	1,067	↑
7 Fedora	1,043	↑
8 Debian	791	↑
9 PMagic	767	↑
10 SLAX	601	↑

» **DistroWatch.com** оценивает популярность дистрибутивов, основываясь на числе посещений страницы каждого дистрибутива. Хотя это не дает представление о действительном числе установок, но показывает, какие дистрибутивы более популярны за определенный промежуток времени.

Сбиваясь в стаю



Linux Format слегка удивился, повстречав в Монреале Луиса Суарес-Поттса. Что делает разработчик офисных приложений на конференции Libre Graphics Meeting?

Ранее в этом году *Linux Format* присутствовал на встрече Libre Graphics в Ecole Polytechnique de Montreal в Канаде, в анонсе которой было заявлено: «по поводу участия: художники и разработчики, приносите ноутбуки и покажите, что вы умеете (и чего пока что не умеете). Организуйте BOF-сессию по поводу своего любимого проекта или функции». В отличие от большинства других конференций, где доминируют специалисты по продажам и маркетингу, главный организатор Луи Дежарден [Louis Desjardins] создал Libre Graphics Meeting как мероприятие для разработчиков: воздух прямо-таки гудел от идей взаимного обогащения открытых проектов – и не только тех, которые в первую очередь связаны с графикой.

Мы поймали Луиса Суарес-Поттса [Louis Suarez-Potts], менеджера по развитию и главу сообщества ведущего открытого настольного приложения, *OpenOffice.org* (www.openoffice.org), с целью узнать его мнение о задачах, стоящих перед многими открытыми проектами, в особенности о реализации Open Document Format [ODF] и о преимуществах и недостатках корпоративного вклада в разработку программ с открытым кодом.

Linux Format: **Чем заполнен ваш рабочий день, кроме вашего обширного и изнурительного вклада в OpenOffice.org?**

Луис Суарес-Поттс: Я являюсь связующим звеном с основной компанией, которая поддерживает и спонсирует *OpenOffice.org* [*OOo*] – Sun Microsystems, а помимо этого с 2000 по 2007 работал старшим менеджером по развитию сообщества и консультантом по стратегии Open Source в CollabNet (www.collab.net), софтверной компании, которая предоставляет хостинг *OpenOffice.org*. Недавно знакомился с проблематикой «инфономики», или политэкономии информации. Некоторые из моих публикаций и работ можно найти на http://homepage.mac.com/luispo/. Один из моих блогов посвящен *OpenOffice.org* и Open Source: http://ooo-speak.blogspot.com, другой – культурному критицизму.

LXF: Второй вопрос может показаться нахальным, но зачем вы приехали на Libre Graphics Meeting? Выбор этой конференции выглядит довольно странным для того, кто так плотно занят более… э-э… Office'ной стороной приложений…

Жаргон

BOF
Сокращение от пословицы «Одной породы птицы в стаю стремятся сбиться» [в оригинале – "Birds of a feather flock together" – *прим. пер.*] в компьютерных кругах обычно называются неформальные дискуссии по интересам, возникающие спонтанно, без предварительного плана.

Nick Veitch

ЗДРАВЫЙ СМЫСЛ БИЗНЕСА

«В мире Open Source вы ничего не теряете, рекламируя чей-то чужой продукт».



»

» **ЛСП:** По двум причинам. Во-первых, это конференция для разработчиков. Большинство участников так или иначе связаны с открытой графикой. И, конечно же, есть открытый формат документов. Хартия ODF по большей части касается документов, но ведь графика тоже используется в презентациях и т.п. Проявлен большой интерес, при большой активности, насчет того, как *OpenOffice* может перерасти рамки традиционного офисного пакета, чтобы включить вещи, весьма интересные с творческой точки зрения, и решить проблему использования ODF в других программах, не входящих в этот пакет. Например, *Scribus*; если вы можете импортировать свой текст в формате *OOo*, почему бы вам также не сохранять документы и не разделять доступ к ним как к документам с открытым кодом, по типу ODF? Ведь преимущество вполне понятно, если, например, вам надо постоянно обновлять страницу, используя данные из электронной таблицы.

LXF: А вторая причина?

ЛСП: Это никак не связано с конкуренцией – никто из нас не конкурирует с каким-то «плохим парнем»; в мире Open Source всем надо держаться вместе. Каким бы открытым проектом ты ни занимался, у тебя есть своя задача, но все наши действия нацелены на одно – предложить конечному пользователю выбор удобного ПО. Само понятие сотрудничества обязано выходить за рамки любого открытого проекта. Общий формат важен, но не менее важны такие вещи, как сходство технологии, архитектуры, кода – все это важно. Никто не собирается покупать такие приложения, как *KOffice* или *OpenOffice.org*, но пользователи могут выбрать одно из приложений с открытым кодом как альтернативу вездесущему *Microsoft Office* в силу не только его функциональности, но также способа взаимодействия с другими программами на их рабочих столах.

LXF: В этом ведь и есть цель Open Source, не так ли? Это – одно из ключевых отличий программ с открытым кодом от проприетарных программ.

ЛСП: Надо думать! Гоббсовский идеал [Томас Гоббс – английский философ, чьи идеи легли в основу политической философии, – прим. пер.] (что неограниченная конкуренция хороша для бизнеса) несовместим с идеалом Open Source; я не говорю, что все разработчики должны вместе трудиться над каждым проектом, это нереально; зато действительно необходимо сотрудничество с другими проектами, умножающее наши силы для достижения общих целей. Вот одна из причин сотрудничества *OOo* со *Scribus*: *Scribus* весьма разборчив относительно шрифтов и требует соответствия спецификациям, значительно более строгим, чем у большинства других приложений. Обе команды обсудили возможность создания библиотеки шрифтов («библиотеки» в старом смысле: репозитория), соответствующей стандартам *Scribus* и в то же время способной работать с другими совместимыми приложениями, например, *OOo*. Это поможет не только *Scribus* и *OOo*, но также и пользователю.

И с моей точки зрения, использование ODF в *Scribus* и в других сходных приложениях весьма желательно, хотя и не столь важно, как усовершенствование функции импортирования. Кроме всего прочего, в Libre Graphics Meeting есть отличная вещь: публикация превосходного сборника. Он показывает, как хорош *Scribus* (Луи Дежарден сам участвует в этом проекте) и насколько серьезны усилия, вложенные в это мероприятие.

LXF: Если расширить функциональность программ с открытым кодом, это утащит большее число потенциальных пользователей от проприетарных программ?

ЛСП: Я ни на секунду не предлагаю *OOo* стать боргом в мире программных продуктов, сжирающим все! [Borg – фантастическая раса киборгов в Star Trek, грубая сила, сопротивляться которой бесполезно, – *прим. пер.*] Отличный пример стратегии Open Source по части взаимодействия и дружелюбия к пользователю – такая простая вещь, как всплывающее окно в *OOo* с советом пользователю обратиться к *Scribus*, если способ форматирования файла пользователем подсказывает программе, что лучше было бы использовать приложение DTP. Точно так же и с графикой – таким же образом пользователям *OOo* будет сообщено о возможности использования *GIMP*, если они попытаются нарисовать в *OOo* нечто более сложное, чем квадрат.

Все это стимулирует пользователей. Многие – если не большинство – пользователей компьютера никогда даже не пытались выяснить, какая программа больше подходит для выполнения их задач: они просто обходятся тем, что есть под рукой и поставлено вместе с компьютером (или с пакетом поддержки, если речь идет о пользователях из крупного бизнеса), вместо того, чтобы оценить другие варианты. В мире Open Source вы ничего не теряете, рекламируя чей-то чужой продукт – хорошо все то, что расширяет сферу пользователей открытого кода.

> ПЕРЕРАСТИ МЫШЛЕНИЕ XX ВЕКА
>
> **«Я ни на секунду не предлагаю OOo стать Боргом среди программ, сжирающим все.»**

LXF: И какую реакцию от других проектов вы получаете, когда говорите о подобном взаимодействии?

ЛСП: Большинству из них это по душе. Очень важно работать вместе над проблемой совместимости. Вчера мы кое с кем из команды *OOo* общались с парнями из *KOffice*, обменивались с ними опытом по усовершенствованию совместимости вообще и говорили о потребностях вокруг ODF. И о том, как лицензировать технологию – весьма важная тема в связи с выходом GPL 3.

LXF: Как вам кажется, не вносит ли участие спонсоров некий оттенок обязаловки для участников подобных конференций?

ЛСП: Все зависит от организации. Это может быть сложно с этической точки зрения: одно дело, когда вас угощает роскошным обедом некая коммерческая группа, выражающая полное одобрение целей Open Source; однако более высокий уровень такого участия может подразумевать определенное коммерческое давление, не всегда уживающееся с академической природой дискуссии. Возможно, для компаний-спонсоров было бы разумнее тратить свои средства не на организацию обедов, а, например, на обучение менеджеров проектов работе в стиле меньшего чинопочитания…

Понятно, что многие компании ощущают наличие некоего конфликта между их бизнес-моделями и целями Open Source; но все же грань между спонсором и участником все более размывается, по сравнению с тем, что было раньше – Novell, можно поспорить, попадает в обе категории: с одной стороны, обеспечивает финансовую поддержку конференций, с другой стороны – одновременно открывает код для сообщества. Должны быть четкие этические

Искусство в открытом коде

Blender
За несколько лет *Blender* прошел путь от закрытого коммерческого приложения до одного из самых обсуждаемых 3D-приложений в мире, и теперь он распространяется свободно. www.blender.org

GIMP
GNU Image Manipulation Program – наиболее широко используемый свободный графический пакет, работающий на разных платформах, включая GNU/Linux, Windows и Mac OS X, и имеющий возможности расширения благодаря модульной архитектуре. www.gimp.org

Inkscape
Лидер среди свободных программ векторной графики с функциями, подобными *Adobe Illustrator*, *Macromedia Freehand* и *CorelDraw*. Используя SVG в качестве родного формата документов, он работает под Linux/Unix, Windows и Mac OS X. www.inkscape.org

Krita
Лауреат премии Академии KDE в номинации Лучшее приложение 2006 года, *Krita* – это приложение для создания и редактирования изображений для *KOffice*, часть офисного пакета KDE начиная с версии 1.4. http://koffice.org/krita/

Scribus
Названный Newsforge «одним из убойных приложений для Linux», *Scribus* представляет собой настольную издательскую систему, которая работает под Linux,Unix, Mac OS X, OS/2 и Windows 2000/XP. www.scribus.net

The Open Clip Art Library
Цель этого проекта – создание архива открытого векторного клип-арта в формате SVG, предоставляемого бесплатно и являющегося достоянием общественности. Содержит более 12 000 работ более 700 художников, доступных для скачивания. www.openclipart.org

границы, однако кто решает, где именно провести эту черту? Большинство корпораций готовы финансировать определенные мероприятия, однако организаторам следует быть осторожными, иначе кто-нибудь задаст скользкий вопрос: «Во сколько обходится заинтересованность разработчика?»

По моему мнению, некоторые компании не выступают в роли спонсоров в прямом смысле этого слова – они готовы вкладывать ресурсы, поставляя людей для работы в областях программирования, лежащих в пределах их собственных разработок, в ущерб прочим областям, и не работают с сообществами, не занимающимися разработкой – то есть с другими компаниями, заинтересованными именно в тех областях Open Source, куда и вкладываются эти спонсорские компании. А те другие стороны в конце концов получают доступ к коду, потому что все возвращается к исходному моменту, но не получают понимания причин, по которым код стал именно таким.



LXF: Значит, между бизнесом и сообществом непреодолимая грань?

ЛСП: Вовсе нет. В случае *OOo*, где-то в 2002-3 г. вопрос совместимости с Mac OS X решался широко распределенной международной группой, состоящей из людей, которые вместе работали над одной задачей – может ли быть лучший пример сообщества? Хорошо бы было побольше таких компаний, как Novell, в полном масштабе работающих с сообществом, а не только нацеленных на освоение рынка. Novell является спонсором конференции *OOo* Con, и это прекрасно, но лучше бы использовать эти деньги не на круизы, а на помощь с контролем качества, локализацию и обучение, и тогда получаемая отдача будет намного лучше и для Novell, и для сообщества в целом.

Люди думают, что это разделение существует из-за самой природы спонсорства или пожертвований разных участников. Например, IBM не может внести большего вклада в развитие *OOo*, потому что *OOo* конкурирует с некоторыми из ее продуктов; или зачем Adobe содействовать открытому проекту, который станет конкурентом его продуктов, поставляющих солидную часть его доходов?

LXF: Сложно ли координировать проект, в котором участвуют как добровольцы, так и оплачиваемые сотрудники?

ЛСП: В классической корпоративной структуре есть отдел маркетинга, который говорит: «Вот что нужно пользователям», потом эта информация синтезируется и объясняется инженерам; потом инженеры говорят: «Вы шутите!» Так вот, в мире Open Source это самое «Вы шутите!» формулируется куда более резко, потому что разработчики-добровольцы могут просто взять и уйти, если у них пропал интерес к делу – они хотят заниматься тем, что представляет для них больший интерес с точки зрения технологии. Это – классический либеральный подход (в смысле свободы, а не политики), и в наши дни открытым кодом по большей части руководят люди, привыкшие работать именно в таком стиле.

Когда компания выделяет определенное количество разработчиков для работы в определенной области, могут возникнуть трения, поскольку разработчики не в восторге от того, что компания распределяет ресурсы или диктует, над какими частями проекта следует работать. Эти области не всегда привлекательны для разработчиков, потому что не дают больших возможностей для инноваций. Многие разработчики не хотят просто изготовлять открытые эквиваленты для функций, уже предоставленных программами компаний вроде Microsoft и Adobe; ими движет желание реализовать что-нибудь потрясающее. Если специалисты по маркетингу не считают это стоящим, или группа изучения спроса изначально не увидит здесь изюминки для конечного пользователя, то коммерческие соображения приведут к ограничению ресурсов, выделенных на развитие новых функций, а для разработчиков все закончится тем, что им придется работать над интересными для них областями исключительно безвозмездно, без всякой спонсорской помощи. Многих разработчиков Open Source – и оплачиваемых, и добровольцев – вероятно, огорчает такая ситуация, но они понимают, что на данный момент дела обстоят именно таким образом, и как-то мирятся с этим.

LXF: Как вы считаете, аспект пиара в разделении кода со всем открытым сообществом является серьезным мотивом для этого?

ЛСП: Я довольно скептически отношусь к тому, как это работает в рыночных отношениях, но свои преимущества у этого определенно есть. Microsoft тратит больше денег на подбор разработчиков, чем, например, Sun. Однако вам не нужны рабы Microsoft'а [в оригинале – Microserfs, дословный перевод – «Рабы Майкрософта», роман английского писателя Дугласа Коупленда, – *прим. пер.*]: как наниматель, вы заинтересованы в поиске разработчиков, предпочитающих технологическую и социальную свободу, которую дает открытый код – пиар играет важную роль в привлечении новаторов и интересных людей; успехи Sun в привлечении множества хороших разработчиков на данный момент очевидны.

В противовес тому, что я говорил раньше о слишком большом влиянии предприятия на проекты, связанные с их моделью извлечения прибыли, будет правильно сказать, что привлечение стремящихся к получению прибыли компаний может дать важное направление открытым проектам. Посмотрите на Debian…

LXF: Debian? Ха-ха!

ЛСП: Можете смеяться! Debian – эксперимент по демократии, пошедшей вразнос, вплоть до амока [амок – приступ неудержимой ярости или безумия, – *прим. пер.*]. И он этим очень гордится: это же так радикально; но подобный разгул демократии может отвратить деловые круги от Open Source, поскольку способен препятствовать прогрессу. Разработка не происходит в вакууме; в бизнесе, чтобы запустить продукт, нужен кто-то, чья ответственность – говорить: «вот в этом направлении мы будем двигаться» и «вот чем мы займемся после!» – потому-то Марк Шаттлворт (руководитель Ubuntu, основанного на Debian) столь важен для Linux.

LXF: Интересно отметить, что, хотя ODF напрямую не поддерживается в *Windows Office 2007*, Microsoft финансирует проект *ODF add-in for Word* на SourceForge для создания модуля расширения *Microsoft Office*, свободно доступного по лицензии BSD. Сколько компаний понимают это?

ЛСП: По мере распространения Open Source, компании начинают понимать, что к чему, и поворачиваться в верном направлении. Стоит им осознать, что больше не придется делать все свои исследования в рамках своей компании, выиграют все – если они начнут финансировать открытый проект, это станет подобно открытию дочернего бизнеса, но только с намного меньшими рисками и во многих случаях куда более окупаемого. LXF

Битва за лучший ubuntu

«Gutsy Gibbon» [«Бесстрашный Гиббон»] почти вышел – удержит ли команда Ubuntu звание популярнейшего в мире дистрибутива? **Майк Сондерс** разберется...

Дистрибутивы приходят и уходят, как времена года. WinLinux, AlphaNet, Xpresso и многие другие возникали на наших радарах, хвастались новыми возможностями и обещали нанести имена своих создателей на скрижали истории Linux. Ну и где они сегодня? Что произошло? Почему разговор сейчас идет только о Fedora, OpenSUSE, Ubuntu и Ко?

Ответ прост: сообщество. Галдеж о так называемых революционных функциях может вызвать пару недель ажиотажа, но не создаст «долгоиграющий» дистрибутив. Так многие из якобы перспективных дистрибутивов, которых мы нагляделись за эти годы, в конце концов ушли в небытие, зачахнув без поддержки разработчиков и участия сообщества.

> «У популярности свои проблемы: прежде всего, нельзя ронять планку.»

Но команда Ubuntu все это получила. Они взяли за основу хорошо отлаженную базу Debian, отшлифовали ее, приспособив к пользователю, и поддерживают ее с помощью исключительно работящего и живого сообщества. Сила Ubuntu в том, что он редко пытается вводить пионерские функции, изменяющие мир Linux – он довольствуется стандартными утилитами конфигурации Gnome и отстает по части вызывающих слюнки украшательств вроде *Compiz*. И тем не менее, это один из самых популярных дистрибутивов в мире.

У популярности есть свои проблемы: прежде всего, нельзя ронять планку. Группа Ubuntu жонглирует обширным массивом потребностей пользователей, решая, какие вещи заслуживают включения в новый релиз, и исправляя столько ошибок, сколько позволяют наличные людские ресурсы. Трудные решения должны приниматься, споры пресекаться – а накопившиеся корзины с грязным бельем вытряхиваться в процессе открытой разработки.

Итак, каким же образом команда Ubuntu и ее родитель – компания Canonical – объединяет все эти элементы и издает новый релиз дистрибутива? На следующих семи страницах мы будем вникать в процесс развития Ubuntu, анализируя предстоящий релиз Gutsy Gibbon (7.10) и прослеживая его прогресс от возникающих идей до исполняемого кода. С помощью ключевых фигур среди хакеров Ubuntu мы поймем, что необходимо для успеха релиза – даже если вы не используете этот дистрибутив, вы окунетесь в самую гущу процессов и людей, собирающих новую версию Linux.







Зарождение

Перед началом пиршества разработки хакеры Ubuntu должны решить, какие функции следует включить.

Представьте себя разработчиком Ubuntu: вы только что открыли дверь последнему, сияющему, новехонькому релизу, и пользователи Linux по всему миру азартно его устанавливают. Вы бросаете написание кода и исправление ошибок, созрев для вполне заслуженного отдыха – но не тут-то было! Уже пора вгрызаться в следующую версию. Ubuntu следует шестимесячному графику разработки (с единственным отклонением в версии 6.06 из-за проблем со стабильностью), поэтому разворот к новым функциям выполняется очень резко.

В команду разработчиков Ubuntu входят и оплачиваемые, и добровольные кодеры: первые трудятся над дистрибутивом полный рабочий день и нанимаются Canonical, а вторые подключаются, когда сами могут. Несмотря на этот «гибрид» платной и бесплатной работы, между двумя группами практически не существует трений, с тех пор как в прошлом Canonical наняла некоторых разработчиков Debian. Для многих кодеров участие в сотворении Ubuntu обеспечивает дальнейшую карьеру в области оплачиваемого создания дистрибутивов.

После отправки нового выпуска Ubuntu на зеркала снова запускается процесс отбора функций для следующей версии. Мэтт Циммерман, технический директор Ubuntu и председатель Технического совета, поясняет: «Идеи предлагаются сообществом Ubuntu и самими разработчиками. Если разработчику нравится идея, он воплощает ее в технической спецификации, которая затем может стать основой для проекта».

Идеи будущих функций накапливаются на *Launchpad*, сайте поддержки разработчиков Canonical – для Gutsy они перечислены на https://launchpad.net/ubuntu/gutsy/+specs. Некоторые идеи обсуждаются разработчиками лично при встречах. Циммерман продолжает: «Основная активность проявляется на саммите разработчиков Ubuntu, перед началом каждого цикла выпуска. Спецификации рассматривает группа технических экспертов для оценки их правильности и жизнеспособности».

Приоритеты! Приоритеты!

Каждой идее новой функции присваивается приоритет (высокий, средний или низкий), зависящий от спроса на нее, ее влияния на общую стабильность дистрибутива и от того, уложится ли ее выполнение в срок. *Launchpad* также постоянно информирует о процессе работы над новой функцией: разработчики могут видеть, находится ли процесс в самом начале, хорошо ли продвинулся или готов к бета-тестированию. Здесь же сообщается, кто работает над каждым проектом.

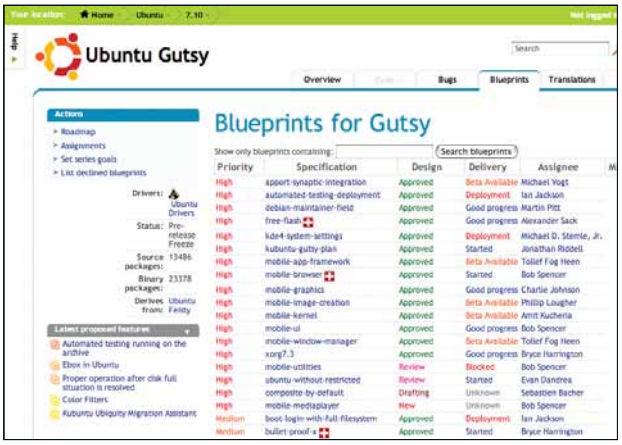
Но как распределить усилия разработчиков? Слово Циммерману: «Canonical определяет, какими проектами займутся ее разработчики, а сообщество само решает, на что тратить свое время. На практике, многие разработчики Canonical занимаются проектами по своим же идеям, но некоторые решения мы принимаем “сверху”». Так что добровольные участники могут облюбовать себе для разработки функции, которые им нравятся, а Canonical придаст определенную направленность своей оплачиваемой команде.

Предложения по функциям поступают со всех углов мира Linux: некоторые дополнения берутся из других дистрибутивов, некоторые зарождающиеся технологии просачиваются из случайных сайтов на SourceForge, а иные являются просто классными примочками, которые пригодятся в следующем релизе Ubuntu. Когда полный список функций готов, менеджер по разработке (в Gutsy это Скотт Джеймс Ремнант [Scott James Remnant]) отправляет его в рассылку разработчиков (http://tinyurl.com/2cxfev), и начинается кодинг...

Мои любимые функции Gutsy: Марк Шаттлворт



«Я разрываюсь между *Compiz*, предоставляющим замечательное многомерное окружение рабочего стола, и новым Ubuntu Mobile для интернет-планшетов, позволяющих вам носить Сеть в кармане.»



› http://launchpad.net **помогает разработчикам в организации и расстановке приоритетов новых функций Gutsy.**

Повышаясь в ранге

Существует два типа разработчиков Ubuntu: MOTU и Core. Первые является частью команды “Masters of the Universe [Хозяева Вселенной] ”, это большая группа хакеров, поддерживающих программы в репозиториях Universe и Multiverse. Universe охватывает пакеты, которые не считаются достаточно важными для базовой системы, но тем не менее востребованы пользователями. Новые разработчики присоединяются к группе MOTU, упаковывающей программы в Deb-пакеты Ubuntu и поддерживающей исправление их ошибок и проблем безопасности. Показав свою способность создавать хорошие пакеты для Universe, разработчик может просить Технический совет Ubuntu и Совет сообщества присоединить его/ее к группе Core. Если разработчик проделал хорошую работу: создал надежные пакеты, имеет сильные коммуникативные навыки и обширные знания Linux, его/ее принимают в группу Core, работающую над критичными пакетами в репозиториях Main и Restricted (т.е. предназначенных для дисков Ubuntu). Ubuntu имеет свой кодекс поведения: разработчики обязаны взаимно оказывать друг другу помощь, уважение и сотрудничество.

Время кодировать

«Создание дистрибутива – это 1% вдохновения и 99% потогонного труда» – так мог бы сказать Томас Эдисон, живи он во времена Linux...

Хотя Gutsy сейчас [на момент написания статьи, – прим. пер.] находится посереди кипучего процесса разработки, общие цели для него уже установлены, как описано выше. Ключевые функции и поправки команда Ubuntu надеется включить в 7.10, но, как всегда бывает, при релизе дистрибутива некоторые из них могут отпасть, если они не заработают или не будут достаточно стабильными на дату выпуска!



› Любуйтесь! 3D-эффекты рабочего стола в действии, благодаря интеграции *Compiz Fusion*.

» **Рабочий стол Gutsy** поставляется с Gnome 2.20, а его Kubuntu-вариант – с KDE 3.5.7. Желающие получить представление о KDE 4 смогут заодно установить его релиз-кандидат. Gutsy впервые включит *Compiz Fusion*, обеспечивающий умопомрачительные 3D-эффекты рабочего стола на поддерживаемых графических чипах. Пользователи смогут вернуться к стандартному менеджеру окон за пару щелчков мыши. Кроме того, в работе находится новый графический инструмент настройки экрана (разрешение, количество цветов и т.д.) на базе *GTK*.

» **X.org X**, оконная система, претерпит несколько изменений: будет включена версия 7.3, обеспечивающая горячее подключение устройств отображения. Одна из преследуемых здесь целей – «пуленепробиваемые X», то есть настройки X'ов при возникновении проблем спокойно возвращаются к стандартным видеорежимам, а не обрушиваются с кучей вывода в командную строку. В интересах всех новичков Linux в мире, мы одобряем эту функцию, с высокой *Башни LXF*!

» **Win-модемы** Да, эти противные программные модемы, наиболее широко распространенные на ноутбуках или розданные бесплатно с пакетами широкополосной связи, будут поддерживаться. По крайней мере, некоторые из них. Gutsy обеспечит доступ к некоторым «ограниченным» (restricted – то есть не совсем свободным или открытым) драйверам, чтобы помочь людям выйти в сеть.

» **Безопасность** AppArmor, система безопасности, позволяющая администраторам блокировать используемые программы и ресурсы, будет доступна как опциональное дополнение. Это система, используемая в Novell SUSE; Red Hat придерживается SELinux.

» **Свободный Flash** Некоторые кодеры Ubuntu изучают возможность включения свободного Flash-плейера. Сейчас наиболее приемлемый вариант – это несвободный плейер от Adobe, но команда Ubuntu присматривается к *Gnash* – свободной альтернативе. Однако, *Gnash* по-прежнему отстает от проприетарного плейера.

» **Свободный как свобода** В техническом обзоре заявлена возможность «не ограниченной» (non-restricted) установки Gutsy, то есть установки с CD без копирования несвободных компонентов на ваш жесткий диск. Считается, что это более разумный подход, чем создание отдельного дистрибутива для сторонников чистоты Open Source и свободного ПО. Он будет называться Gobuntu.

» **Варианты** Разработчикам, занимающимся интернет-планшетами и другими небольшими устройствами, будут предложен новый вариант Ubuntu: Mobile and Embedded Edition. Кроме того, проект Ubuntu Server получит новую команду от Canonical.

Таковы текущие цели особенностей Gutsy, и при нынешних темпах развития уже видится, что большинство из них войдут в окончательный релиз. На этой стадии у ведущих разработчиков начинают шалить нервы – ведь сайты новостей и журнальные статьи (одна из них перед вами!) принимаются обсуждать новые возможности, и неприятно думать о разочарованиях, последующих, если при окончательном раскрое релиза выкинется что-то важное. Циммерман описывает эту проблему как интенсификацию разработки: «Главное – чтобы множество параллельных проектов шли голова к голове и мы успели бы сделать все запланированное».

Кодировочная лихорадка

И пошло кодирование! Сборщики пакетов из команд Core и MOTU хватают новейший исходный код тысяч программ, добавляют скрипты компиляции Deb-пакетов и генерируют их в духе Ubuntu. Дистрибутив в это время подобен водовороту, постоянно бурлящему и меняющемуся под влиянием разработчиков и перестроения критических зависимостей (например, библиотек Gnome и KDE), поэтому сборщикам

Ключевые фигуры Ubuntu

Уважаемые люди в команде хакеров Ubuntu:

» **Марк Шаттлворт** [Mark Shuttleworth] Мультимиллионер, совершивший вылазку в космос; сделал себе состояние, основав Thawte, а теперь он самопровозглашенный «Пожизненный Добрый Диктатор». Он задает общее направление проекта и, как бывший разработчик Debian, также вносит свой вклад в некоторые технические решения.

» **Мэтт Циммерман** [Matt Zimmerman] Технический директор. Он заботится о коде дистрибутива, удостоверяясь, что процессам разработки и выпуска не препятствуют крупные ошибки, прения разработчиков или чрезмерные запросы функций.

» **Брайан Мэррей** [Brian Murray] Главный спец по ошибкам и менеджер команды Ubuntu QA, ответственной за то, чтобы «задавить» побольше ошибок в дистрибутиве перед релизом.

» **Кори Бургер** [Corey Burger] Глава маркетинговой команды; отвечает за контакт с группами СМИ по информации о релизах и обеспечивает, чтобы они излучали положительные волны.

» **Трой Соботка** [Troy Sobotka] Лидер группы художников. Его команда работает над темами и иконками, чтобы Ubuntu выглядел свежо (ну, то есть, так свежо, как может выглядеть коричневая тема) и гарантирует, что дистрибутив визуально гармоничен.

» **Мэтью Ист** [Matthew East] Наблюдает за документацией, которая помогает делать дистрибутив более доступным, и обеспечивает переводы на множество языков.



нужно постоянно следить за ним и убеждаться, что их родные проекты синхронизированы. Допустим, ответственный за сборку графического инструментария *GTK* собрал новый Deb-пакет со свежей версией; тогда сборщики *GTK*-приложений должны гарантировать, что их продукты будут с ним совместимы и запускаться без проблем.

Хотя Ubuntu и построен из пакетов, часть из них – не просто отдельные программы: они существенно преобразуют систему. Например, скрипты запуска (инициализации) существуют в Deb-форме, но их содержание и действия могут внести радикальные изменения в работу Ubuntu. То же относится к пакетам ядра, способным включить или отключить критические части системы (например, драйверы оборудования). Как объясняет Циммерман, такие изменения тщательно проверяются: «Существенные или спорные изменения обычно неофициально обсуждаются сообществом, и проблема решается на основе консенсуса. Если консенсус не достигается, окончательное решение выносит Технический совет Ubuntu».

Разработчики постоянно выгружают свои пакеты с новыми версиями программ или исправлений, и вся эта работа открыта для желающих поучаствовать. Многие рьяные фанаты Ubuntu стремятся «догнать ветвь разработки» – это жаргонное выражение для постоянного обновления до самых последних версий пакетов. Занятие рискованное, но энергичные пользователи отлавливают ошибки и помогают с ними покончить. Возможно, вы слышали термин «триаж» в применении к ликвидации ошибок, но не знаете, что он означает. Когда пользователи представляют отчеты об ошибках в Ubuntu, то ошибки идут на триаж: определяется степень их серьезности и что нужно с ними делать.

Но есть одно «но». Большинство разработчиков Ubuntu заняты превращением в пакеты программ других хакеров, будь то небольшой консольный почтовый клиент или пакет-гигант типа *OpenOffice.org*. Так что группа Ubuntu должна находить баланс между работой с «притекающими сверху» (от разработчика) исходными кодами и целями дистрибутива. Циммерман пишет: «Труднее всего, когда проект связан с деятельностью, протекающей вне Ubuntu; мы часто зависим от своевременного окончания работ в сторонних открытых проектах.» Если разработчик программы испортил ее и добавил вагон ошибок, сборщик пакетов Ubuntu должен решить, принять ли новую версию или оставить старую (и более устойчивую), рискуя нарваться на протест жадных до новинок пользователей. А иногда, наоборот, сборщики Ubuntu отчаянно ждут новых функций, а таймер «релиз-каждые-шесть-месяцев» тикает у них в ушах.

Мои любимые функции Gutsy: Даниэль Холбах



«*Compiz*. Я уже несколько недель его использую, и это изумительно, как плавная анимация и оконные эффекты создают абсолютно новый стиль и характер работы. Также ожидаю множества улучшений и хороших вещей в Ubuntu Server (*AppArmor*, *EBox* и т.д.) и просто замечательный Ubuntu Mobile.»

Сборка пакетов и тестирование

Процесс сборки пакетов с новым ПО, исправления ошибок и добавления изменений в базовую систему продолжается от одного до двух месяцев, прежде чем команда сможет показать что-то конкретное миру. Каждый пользователь Ubuntu может скачать последние версии пакетов и протестировать их на любой стадии, но для упрощения ситуации в более широком мире пользователей Linux, команда Ubuntu выкладывает текущий вариант дистрибутива, отражающий прогресс в работе над ним. Как и окончательный релиз, это автономный, инсталлируемый CD, но он предназначен только для разработчиков – хакеры Ubuntu отговаривают случайных пользователей от его применения.

Каждый дистрибутив имеет кодовое имя для серии таких вариатов: в Feisty Fawn (7.04) они назывались “Herd” [Стадо], а нынешний Gutsy использует кодовое имя “Tribe”[Племя], так что в процессе разработки команда плодит Tribe 1, Tribe 2..., демонстрируя новые возможности и собирая отчеты об ошибках. Эта обратная связь жизненно важна для дистрибутива: она позволяет определить, с чем все хорошо, что требует доработки и что следует отставить ради соблюдения сроков.

Затем, примерно за шесть недель до окончательного релиза, безудержная разработка останавливается: новые функции уже не добавишь, радикальные изменения делать некогда. Пришла пора набрасываться на ошибки и делать релиз-кандидаты...



Creative Commons image, RAMONRAMON, http://tinyurl.com/2rz7v5

› Хакеры Ubuntu генерируют идеи и коды на саммите разработчиков в Севилье.

Цикл выпуска: временная шкала

МЕСЯЦ 0

» Наброски новых идей

» Постановка задачи по функциям

» Осуществление разработки

Выпуски текущих снимков

Выходы снимков (snapshots), основная часть графика разработки, как правило, происходят примерно раз в три недели. События здесь планируются в начале выпуска, но точные даты могут меняться из-за ошибок или изменения набора функций.

» Релиз-кандидаты

» Финальная отправка на зеркала

» Анонс релиза

МЕСЯЦ 6

Собери себе Ubuntu

Вот вам способ создать свой индивидуальный вариант Ubuntu, включающий только нужные вам пакеты – так же, как мы делаем это для DVD Linux Format.

Не так уж трудно пересобрать Ubuntu, если вы понимаете принцип работы дистрибутива и не боитесь командной строки. Создание индивидуального варианта – это также отличный способ прочувствовать процесс сборки дистрибутива: вы поймете все сложности, обычно скрытые в релизе Ubuntu за завесой Gnome. Нижеследующее руководство покажет, как извлечь Ubuntu CD, добавить и удалить необходимые вам пакеты, а затем вновь собрать его воедино. Это один из методов, который мы используем для создания супер-мега-расширенной версии Ubuntu на LXF DVD (например, в LXF94 и LXF88)! Имейте в виду, что вам потребуется по крайней мере 3 ГБ свободного места на диске.

1 Получение ISO-файла

Для пересборки Ubuntu нужен образ Desktop CD (тот, что запускается в Live-режиме, а не Alternate CD на базе командной строки). Вы можете скачать последнюю стабильную версию с http://releases.ubuntu.com/feisty/. Скачайте образ **ubuntu-7.04-desktop-i386.iso**, запишите его в свою домашнюю директорию и переименуйте в **feisty.iso**. (Это добрых 700 МБ, и тем, кто не обладает терпением святого, здесь не обойтись без широкополосного интернет-соединения!)

2 Монтирование образа

Образ CD нужно подмонтировать к файловой системе. Откройте терминал и, переключившись на администратора (**sudo bash**), введите следующее:

```
mkdir /mnt/loop
mount -o loop feisty.iso /mnt/loop
```

Теперь содержимое **feisty.iso** доступно в **/mnt/loop**. Нам надо скопировать его в нашу файловую систему, так что создайте директорию и скопируйте все содержимое, вот так:

```
mkdir ubuntu-rebuild
rsync -ax /mnt/loop/. ubuntu-rebuild
```

После этого директория **ubuntu-rebuild** будет содержать файлы с диска. Теперь можно отмонтировать ISO-образ (**umount /mnt/loop**).

3 Подготовка

Далее необходимо распаковать сжатую файловую систему с CD Ubuntu; для этого используем *SquashFS*, представленной в Ubuntu в пакете *squashfs*; вам также необходимо установить *squashfs-tools*. Сжатая файловая система подмонтируется так:

```
mount ubuntu-rebuild/casper/filesystem.squashfs /mnt/
loop -t squashfs -o loop
```

Теперь в **/mnt/loop** находится содержимое сжатой файловой системы Ubuntu – той, что используется, когда вы загружаетесь в режиме LiveCD. Скопируйте ее в новый каталог вашей домашней директории:

```
mkdir ubuntu-source
rsync -av /mnt/loop/. ubuntu-source
umount /mnt/loop
```

4 Изменение пакетов

Ура: теперь все на своих местах! Теперь немного поколдуем для переключения на файлы дистрибутива, как будто именно его мы и запустили. За это надо сказать спасибо маленькому инструменту *chroot*: он меняет восприятие корневой файловой системы, и можно «притвориться», что вы находитесь в другом дистрибутиве. Введите следующие команды – первая из них настраивает сетевое соединение внутри файловой системы Ububntu:

```
cp /etc/resolv.conf ubuntu-source/etc/
chroot ubuntu-source
```

Теперь вы находитесь внутри каталога **ubuntu-source**, как если бы он был корневым (**/**). Вы запустили тот же дистрибутив, который идет на Live CD, но теперь его можно изменять! Я не рекомендую удалять пакеты, если вы абсолютно точно не уверены, что делаете; некоторые являются зависимостями для критичных системных пакетов. Но вы можете начать добавление программ с помощью *apt-get* – например, чтобы добавить *AbiWord* в вашу систему:

```
apt-get install abiword
```

Программы, добавленные на этой стадии, будут присутствовать на финальном CD/DVD, когда мы его пересоберем. Так что вы можете добавить *Xfce*, если это ваше любимое окружение рабочего стола, или пакет ''build-essential'', если вам нужен GCC и иже с ним, готовое к запуску.

5 Обновление

Закончив, наберите **exit**, чтобы покинуть файловую систему Ubuntu и вернутся в свой обычный дистрибутив. Теперь надо сгенерировать список файлов, имеющихся в обновленном образе Ubuntu. Для этого наберите следующую команду-монстр:

```
chroot ubuntu-source dpkg-query -W --
showformat='${Package} ${Version}\n' | grep -v deinstall
> ubuntu-rebuild/casper/filesystem.manifest
```

Необходимо также сказать установщику Ubuntu, когда он запустится, чтоб игнорировал определенные пакеты. Например, после инсталяции иконка «Установить Ubuntu» [Install Ubuntu] на рабочем столе вам уже ни к чему! Итак, пробежимся по списку инсталлируемых файлов (при помощи *sed*) и удалим компоненты исключительно для Live CD:

```
cat > /tmp/sedscript <<END
/casper/d
/libdebian-installer4/d
/os-prober/d
/ubiquity/d
/ubuntu-live/d
/user-setup/d
END
sed -f /tmp/sedscript < ubuntu-rebuild/casper/
filesystem.manifest > ubuntu-rebuild/casper/filesystem.
manifest-desktop
```

6 Ремастеринг

Теперь нам надо пересобрать образ файловой системы *SquashFS* – это займет до получаса...

```
mksquashfs ubuntu-source/ ubuntu-rebuild/casper/
filesystem.squashfs -noappend
```

Далее обновим контрольную сумму MD5 для подтверждения целостности диска:

```
(cd ubuntu-rebuild && find . -type f -print0 | xargs -0
md5sum > md5sum.txt)
```

И наконец, команда **mkisofs** создат нам сияющеновый ISO-образ:

```
mkisofs -r -V "My Modified Ubuntu" -cache-inodes -J -l
-b isolinux/isolinux.bin -c isolinux/boot.cat -no-emul-
boot -boot-load-size 4 -boot-info-table -o newbuntu.iso
ubuntu-rebuild
```

Et voila: Если все прошло «на ура», то вы теперь имеете файл **newbuntu.iso**, который можно прожечь на CD-R (или, если он больше 700 МБ, на DVD). Но помните, что если образ больше 2 ГБ, у вас будут проблемы с запуском. Если же он меньшего размера, то записывайте его, загружайте и наслаждайтесь!

Помощь собратьям

Вы не фанат Ubuntu в режиме 24x7, но стремитесь помочь вашему любимому дистрибутиву? Как правило, большинство проектов ищет кодеров, сборщиков пакетов, тестеров и людей для работы с документацией, как описано на главной странице Ubuntu – применимы те же самые требования. Разберитесь с сообществом вашего дистрибутива и ключевыми игроками, просмотрите списки рассылки, чтобы измерить уровень деятельности и природу отправлений, затем начинайте выдвигать ваши собственные функции или идеи.

- **OpenSUSE**: http://en.opensuse.org/How_to_Participate
- **Fedora**: http://fedoraproject.org/wiki/Join
- **Debian**: www.debian.org/intro/help
- **Mandriva**: www.mandriva.com/en/community/contribute
- **Gentoo**: www.gentoo.org/proj/en/devrel/staffing-needs/



Время R!

Код написан, функции добавлены, и срок выхода релиза все ближе и ближе...

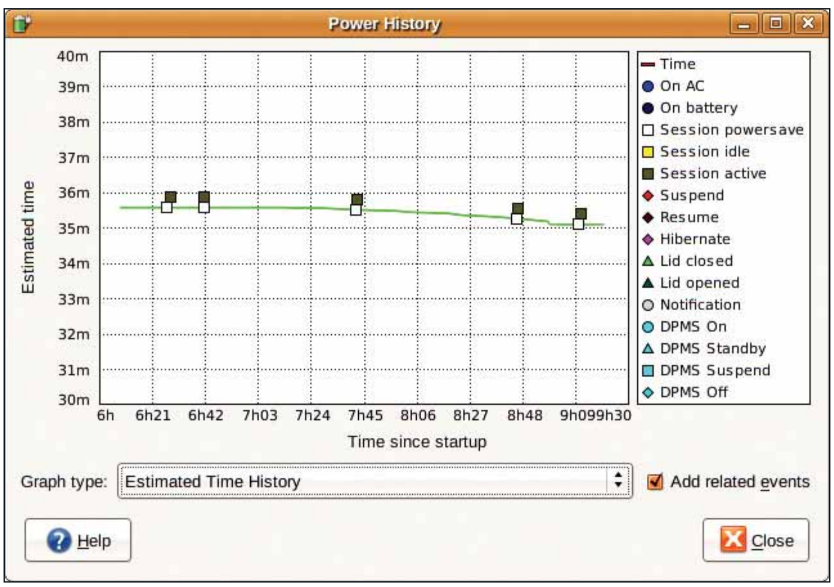
К концу цикла разработка вступает в суматошный период: бурно разрастается количество отчетов об ошибках. Брайан Мэррей руководит Группой контроля качества Ubuntu и управляет уровнем обратной связи с сообществом: «При полностью открытой системе отчетности об ошибках, приятой в Ubuntu, кажется, что мы получаем больше всего отчетов об ошибках, когда уже сделана финальная версия для более широкого распространения. Однако Desktop CD с его Live-средой обеспечивает людям отличный способ проверки релизов для разработчиков и способствует их вкладу в процесс».

Итак, это ситуация «Уловки-22»: команда Ubuntu хочет иметь отчеты об ошибках, чтобы суперски отполировать окончательный релиз, но получает наибольшее их количество тогда, когда дистрибутив фактически уже выпущен! Однако если налицо проблемы со стабильностью, даты выпуска снимков и окончательного релиза могут быть изменены: «Важнее представить нашим пользователям хорошую работу, чем уложиться в назначенный срок». Так было с Ubuntu 6.04 – для устранения некоторых неприятных проблем со стабильностью команда вынуждена была сдвинуть релиз на два месяца ради дополнительной доводки. Поэтому он вышел под номером 6.06.

Недель за шесть до релиза раздается команда: «Заморозить версии пакетов»: никакие новые версии, как от индивидуальных разработчиков программ, так и существующих пакетов Debian, в дистрибутив больше не добавляются. Так, если вы – сборщик пакета *AbiWord* версии 2.4.6 для Gutsy, и тут выходит версия 2.5.0, придется отложить ее до следующего выпуска Ubuntu. Разработчики все еще могут досылать исправления ошибок и проблем безопасности, но цель этого этапа – создание конкретной платформы, которую группа может начать стабилизировать.

Примерно месяц спустя в силу вступает команда «Заморозить функции» – с этого момента никакие новые функции, меняющие работу дистрибутива, добавлять нельзя. Как и при заморозке номера версии, это существенно для сохранения целостности дистрибутива: кому нужно, чтобы какой-нибудь разработчик полностью переписал скрипт начальной загрузки накануне выпуска! Если какая-то планируемая функция не готова к этому времени, она переносится на следующий релиз. В редких случаях разработчик может упросить менеджеров Ubuntu сделать исключение из правила, если выгоды, которые оно принесет, перевесят возможности серьезных последствий по стабильности. Разработчик должен четко продемонстрировать, что его/ее новый пакет не повредит другим, зависящим от него пакетам.

› В Gutsy будет новый измеритель энергопотребления с графиком зависимости этой величины от времени.



Мои любимые функции Gutsy: Кори Бургер



«Я надеюсь, что будут доделаны все работы над сервером. Нужно еще много сделать, чтобы Ubuntu Server был так же «вылизан» и был таким же простым в использовании, как настольная версия. Ведется работа по библиотекам LDAP и сборке *Ebox*, web-инструмента контроля сетевых сервисов.»

Конец не за горами

Далее следуют еще две «заморозки»: художественных работ и «строк». Вторая запрещает изменение текста в программах, а обе вместе они означают, что люди, отвечающие за документацию, могут изготовлять экранные снимки и тексты справок, не опасаясь, что вид дистрибутива внезапно изменится. Через неделю после заморозки строк и за месяц до выхода конечной версии выпускается бета-версия, которая содержит работающие – хотя все еще потенциально нестабильно – реализации всех новшеств дистрибутива. Ubuntu-овщики называют это «функционально-завершенным релизом», подразумевая, что все это пойдет в конечную версию, но нуждается в тестах.

На этом этапе группа исправления ошибок работает на износ, как объясняет Мэррей: «Поскольку близок окончательный релиз, все больше народу берет его [дистрибутив] для прогона, и число сообщений об ошибках может подскочить. Задача здесь – обеспечить быструю сортировку ошибок, определение высокоприоритетных и их устранение».

Помимо технических вопросов, часть сообщества начинает работать над имиджем дистрибутива. Нужно готовить пресс-релизы и стряпать описание релиза с инструкцией по новым функциям, сдобренной экранными снимками. Маркетингом в Ubuntu занимается Кори Бургер: «Команда маркетинга и Canonical обычно беседуют друг с другом незадолго до релиза, чтобы понять, кто что делает. Обычно я звоню одному из маркетологов Canonical и узнаю, чем они занимаются. В Feisty было так, что Canonical занималась пресс-релизом и ответами на запросы прессы (по крайней мере, английской), а сообщество создало множество онлайн-описаний, таких как “Тур по функциям 7.04”».

За неделю до финального релиза команда выгружает сборку RC (релиз-кандидата) дистрибутива: потенциально это окончательный вариант, если в последний момент не найдутся ошибки типа «полный назад». Разработка прекращается; на данном этапе разрешено исправление только критических ошибок. Большинство членов сообщества с нетерпением ждут, надеясь, что их жесткий диск не спалится ни при каких экзотических комбинациях оборудования. Если все хорошо, команда производит окончательную сборку пакетов и ISO-образов дисков, направляет их на зеркала во всем мире и объявляет о новом релизе на сайте Ubuntu.

Тем временем часть разработчиков начинает собирать воедино планы для следующего выпуска Ubuntu. Цикл начинается заново…

Включаемся

Теперь вы получили некоторое представление о процессе разработки Ubuntu; почему бы немного не помочь ему – или любому другому дистрибутиву?

Так как Ubuntu продолжает процветать, проект постоянно ищет новые дарования – и в этом легко принять участие. Не надо быть кодером с большим стажем работы, чтобы помочь дистрибутиву: существует масса других способов влиться в него и добавить свои улучшения. Новый выпуск дистрибутива – это продукт кооперации различных талантов, работающих вместе с одной целью: сделать пригодную к использованию операционную систему. Поэтому, даже если какой-либо из аспектов процесса разработки вам чужд или пугает вас, не бойтесь: можно работать над собственным проектом и больше никуда не соваться.

Как мы видели, решения высокого уровня, определяющие направление развития Ubuntu, принимает Canonical, но подавляющее большинство разработок у всех на глазах. Помощь дистрибутиву сама по себе награда: она дарит вам опыт на будущее. Вот несколько областей, достойных приложения сил (это справедливо и для других дистрибутивов – например, Fedora или OpenSUSE):

» **Кодирование** Создание пакетов, исправление ошибок и выпуск заплаток безопасности являются основой работ в любом дистрибутиве; хотя они не требуют расширенных знаний в программировании. Полезны базовые знания C/C++ и *Bash*-скриптинга. Если вы накоротке с командной строкой, редактируете текстовые файлы и собираете программное обеспечение из исходников, вам по зубам изготовление Deb-пакетов Ubuntu; рассмотрите возможность работы с ними в репозитории Universe. Если вы дока в программировании, попытайте удачи в одном из оплачиваемых bounty-проектов Ubuntu.

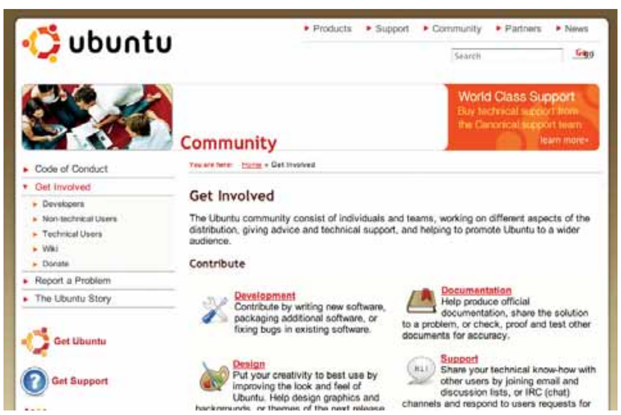
» **Тестирование** Скачайте промежуточный вариант дистрибутива и новые пакеты, и если они не работают, отправьте отчет об ошибках на http://bugs.launchpad.net. Ваши отчеты должны быть краткими, но информативными; слова «Не работает, просьба устранить» только раздражают разработчиков, а точное описание ошибки и спецификация вашей системы поможет им локализовать проблему.

» **Документирование** Одна из проблем Linux – независимо от дистрибутива – это разрозненная документация. Кодеры любят программировать, но их не прельщает перспектива написания руководств пользователя. Вы можете помочь в написании инструкций к еще не документированным программам или улучшить руководства пользователя хорошо известных приложений, имеющих только их черновые наброски. Здесь необходимы знания XML и DocBook.

» **Внешний вид** Как и руководствами для пользователей, этой частью хакеры иногда пренебрегают, что приводит к уродливым или плохо разработанным интерфейсам. Создайте значки, картинки, обои рабочего стола и макеты интерфейса, чтобы ваш дистрибутив выглядел приятнее и понятнее. Еще лучше, если вы сделаете эту графику с помощью приложений с открытым исходным кодом, вроде *GIMP* и *Inkscape*!

» **Поддержка** Сообщество Ubuntu, как и Gentoo, считается одним из самых гостеприимных и дружелюбных; помогите людям на www.ubuntuforums.org, давая советы по основным проблемам пользователей и стараясь, чтобы никто не чувствовал себя чужаком, переключившись на Linux. Пусть вас бесят вопросы вида «пачиМУ линух Ни раБотаит а» – сохраняйте дружелюбие и доступность, и вы заработаете большой авторитет в сообществе.

» **Локализация** Как подчеркнул Ладислав в DistroWatch этого месяца на стр. 20, во многих странах, из-за необходимости знать второй язык, значительная часть населения лишена преимуществ пользования компьютерами. Если вы знаете два и более языка, особенно не поддерживаемых проприетарными ОС, ваша помощь может быть неоценима в любой области, особенно по части поддержки, перевода существующей документации и консультаций разработчикам GUI-интерфейсов.



› **Кодеры, тестеры, художники, составители документации – множество навыков необходимо для создания достойного релиза Ubuntu.**

Мои любимые функции Gutsy: Мэтт Цимерман



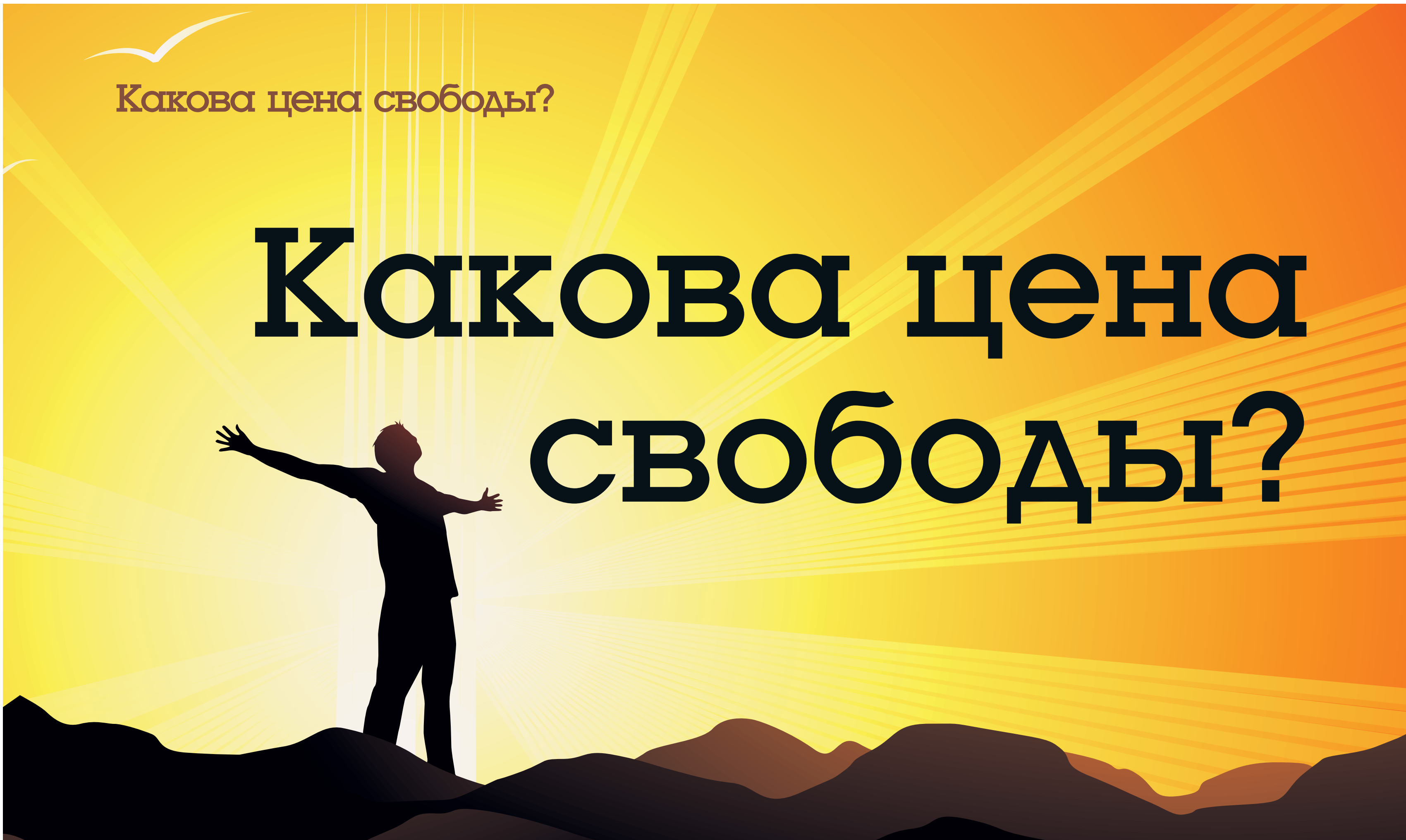
«Я персонально слежу за запуском нашего мобильного продукта – Ubuntu Mobile and Embedded, который будет представлен как часть релиза 7.10. Более подробно об этом можно узнать на сайте http://wiki.ubuntu.com/MobileAndEmbedded.»

Если вы хотите с головой нырнуть в процесс разработки Ubuntu, надеясь на работу в командах MOTU и Core, как было описано выше, вы должны попасть в списки рассылки (https://lists.ubuntu.com). Предложение идей – хорошая штука, но, как однажды сказал Линус Торвальдс, «Разговоры стоят дешево. Покажите мне код». Если вы способны сделать пакет с продуктом, которого пока нет в репозитории Ubuntu, и изготавливаете стабильные, чистые Deb-файлы, то сможете наладить контакт с разработчиками и включиться в списки. Кроме того, если у вас есть идея для базового дистрибутива Ubuntu и вы способны реализовать ее четким, безопасным и легко проверяемым образом, закодируйте ее и разместите в списках.

Ну что, пишем?

Если программирование не ваш конек, но вы хотите приложить руки к улучшению документации дистрибутива, посетите http://doc.ubuntu.com. При склонности к дизайну и графике, попробуйте создавать иконки, логотипы и фоны рабочего стола на https://wiki.ubuntu.com/Artwork. Независимо от того, что вы хотите делать, страница приглашения в Ubuntu на www.ubuntu.com/community/participate предоставит вам ссылки на все виды ресурсов – списки рассылки, страницы wiki и раздел *Launchpad*. Если вы дружелюбны и подробно опишете свою работу или предложение, то сможете получить полезные отклики и множество путей для дальнейшего продвижения.

Или, почему бы не приложить руки к вашему любимому дистрибутиву? Смотрите врезку внизу стр. 31 – там приведены адреса. Удачи! LXF



Какова цена свободы?

В сфере открытого ПО часто приходится идти на компромиссы. **Грэм Моррисон** рассмотрит преимущества и недостатки открытой модели разработки, выбор, который встает перед многими из нас, и его влияние на нашу любимую ОС Linux.

Без обиняков: я люблю *MythTV*. За четыре года оно наполнило медиаконтентом всю нашу жизнь. Мы смотрели фильмы, записывали цифровое и аналоговое ТВ, проигрывали музыку и играли в игры на том самом блестящем черном Linux-компьютере, что спрятан за телевизором.

Компьютер не отличался особенной красотой или бесшумностью, а если вы на какое-то время оставляли дверь закрытой, температура внутри комнаты поднималась до уровня ливийской пустыни. Растения начинали жухнуть, воздух наполнялся кислотой, а уцелевшие граммы влаги скоро испарялись с поверхности раскаленного медного теплоотвода процессора. Пару раз у нас в этой комнате ночевали на раскладушке гости, и если я забывал выключить *MythTV* на ночь, они просыпались с мутными глазами и иссохшие от жажды. Я всегда думал, что небольшой набор таких компьютеров был бы идеальным средством для поддержания температуры в зимнем саду, но этот проект все время откладывался. Вот такой *MythTV*. Сотни часов (я не преувеличиваю) головной боли теперь позади, и перед нами сияющий новый цифровой видеомагнитофон. Я нажимаю на «паузу», и телевидение останавливается. Я нажимаю «запись», и он начинает записывать. Я нажимаю «перемотать», и перемотка заканчивается в считанные секунды. Изумительно. Кроме включения в розетку, ничего больше делать не нужно, и это смущает. С *MythTV* требовалось как минимум потратить выходные на борьбу с командной строкой и скачивание электронного программного руководства, не говоря уж о самой записи.

> «Как пользователи Linux, многие из нас горой за открытые стандарты и свободное ПО.»

Причины радоваться

Есть три причины, почему я сопротивлялся коммерческому решению так долго. Во-первых, цена: я не хотел связываться с услугой, которая по контракту приковывала меня на 12 месяцев. Во-вторых, мне не нравилась компания, которая предлагала эту услугу: она являлась монополистом (поэтому у нас не было альтернативного провайдера), а отсутствие конкуренции заметно усложняет существование независимым проектам типа *MythTV*. В-третьих, я хотел полного контроля над данными, как в *MythTV*. Мы в Linux сейчас принимаем это как должное – мы все используем открытые форматы, и мало кто сталкивается со злоупотреблением запретительными цифровыми правами на компьютере с Linux. Но этого нельзя сказать о многих проприетарных и готовых продуктах. Например, вы не сможете купить MP3-плейер без подобных проблем. И то же самое в мире цифрового телевидения.

Как пользователи Linux, многие из нас стоят горой за открытые стандарты и свободное программное обеспечение. Но это не должно быть первопричиной, ведь Linux – великолепная операционная система, независимо от политики, которой руководствуются разработчики. Многие из нас используют Linux по тем же мотивам, которые когда-то привели меня к *MythTV*: цена, объект финансовых вливаний и контроль над информацией. Соблюдать такую политику нелегко, потому что пользователи Linux часто являются технофилами – они любят суперсовременные примочки и последние достижения техники. Игры, наверное, самый лучший пример. Что вам делать, если вы хотите играть в игры? При желании сохранить верность Linux, вам придется признать, что ваши знания в сфере игр быстро устареют. Средняя Linux-игра продавалась в магазинах под другие платформы еще несколько лет назад, и мы еще рады, что разрыв хотя бы не так велик. Если вы любитель поиграть, остается идти на компромисс: найти себе другое занятие или нарушать одно из трех правил.

»





Игры

Для желающих играть в игры сегодняшнего дня, наша платформа – одна из самых ужасных...

Есть игра?

Небольшие, но затягивающие игры на XBox Live (например, *Space Giraffe* Джеффа Минтера [Jeff Minter]) и на PlayStation недороги, так как затраты на их разработку и распространению ничтожно малы. Возможно, для Linux нужно что-то вроде www. steampowered. com: простая в использовании рыночная площадь для игр от независимых разработчиков. Эта идея может казаться плевком в лицо сторонникам Open Source, но идеи свободного ПО в области игр не столь эффективны, как в области ОС. Деньги в этом случае могут послужить мотивацией.

Wine и подобные программы – прекрасная группа проектов. Они позволяют пользователям Linux играть в игры для Windows, не имея установленной копии Windows на компьютере и без перезапуска машины под Windows. Производительность похуже, чем в исходных версиях, но достаточно близка к ним, а технически это огромное достижение.

Впечатляющим достижением является реинжиниринг библиотек Microsoft DirectX при сохранении двоичной совместимости, особенно при наличии одного-единственного пособия – API программиста. Но в то же время этот подход – своего рода компромисс. Microsoft DirectX высоко ценится для быстрого, мощного развития игр, но он появился в то время, когда многие игры использовали OpenGL – открытую платформу, которая позволила бы заметно упростить перенос игр под Linux. Беда в том, что OpenGL спихнули на обочину, так как многие разработчики перешли на бурно развивающийся эквивалент от Microsoft, отложив разработки для ОС третьих сторон, типа Linux, в долгий ящик. Я не хочу вдаваться в политику по поводу деятельности Microsoft, но если вы хотите, чтоб игры перебирались в Linux, то поддерживать разработку DirectX неправильно. *Wine*, *Cedega* и *CrossOver* – все они добавляют законности к подходу Microsoft, молчаливо признавая, что единственный способ добиться игры в действии – подладиться к этой популярной альтернативе. Linux преследует движущуюся цель: ко времени, когда мы выходим на поле боя, все, что там остается – это спесь по поводу нашего очередного поражения. А то, что действительно нужно сделать – и что мы в некоторых случаях и делаем – это открывать второй фронт, выгодный для наших сильных сторон, а не для слабых.



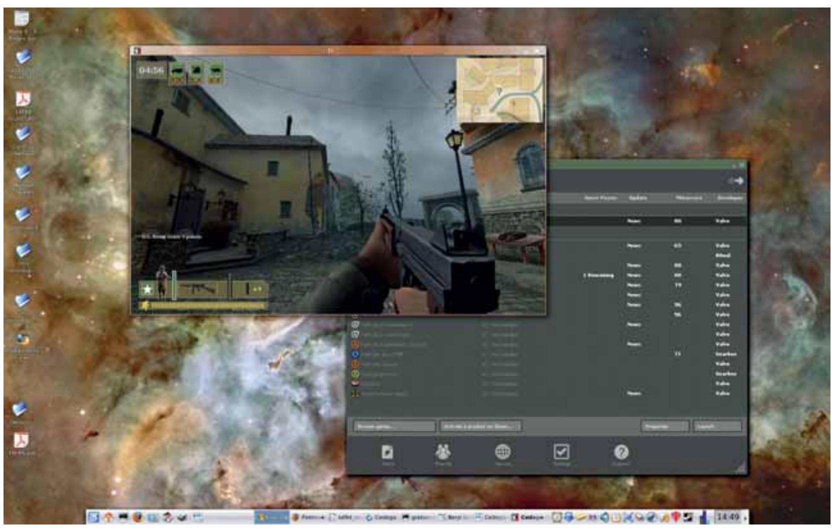
Родные версии игр обычно появляются много позже, чем первоначальный выпуск. _Cold War_ была выпущена довольно быстро – лишь год спустя.

Консоли как терминалы

Игры ставят нас перед другой головоломкой на консольном рынке. Сейчас существуют три основных типа консолей для игр так называемого «седьмого поколения»: выбирать предлагается из Nintendo Wii, Sony Playstation 3 и Microsoft XBox 360. Каждая из них имеет свои преимущества и недостатки, но в настоящее время 360 от Microsoft и Nintendo Wii недалеко ушли друг от друга в борьбе за господство на рынке. Все мы любим Wii, но когда речь заходит о следующем поколении высококачественных игр, мы на самом деле должны выбирать между XBox 360 и Sony PS3. Но почему мы должны делать выбор из этих двух по причинам морального свойства? Почему просто не купить игру, которая лучше, без оглядки на законы рынка и экономические теории? Можно даже сказать, что покупка любой консоли отразится на каждом конкретном производителе, поскольку они понесут потери на каждой единице продаж – таково их желание доминировать. Но чтобы это работало, нужно также отказаться от покупки каких-либо игр в принципе, что поставит крест на предмете покупки, если только вы не покупаете просто плейер для Blu-Ray.

Это аналогично решению, пользоваться *Wine* или нет. Вы стали бы продвигать бизнес-модель, с которой вы не согласны, только ради возможности поиграть? Тут нет правильного или неправильного ответа. Любой из нас с радостью урвал бы Xbox 360, имей он хотя бы полоправдания для покупки. И репутация Sony не намного лучше, даже не считая судебных исков против компании по поводу самовольно установленного миллиона руткитов. Но, по крайней мере, на PS3 можно установить Linux, а в 360 до этого ох как далеко, даже если вы стерпите жуткие надругательства над цифровыми правами. Возможно, мы все должны ждать расцвета рынка подержанной продукции, чтобы выбрать консоль, уже набитую играми.

Тормозит ли _Cedega_ легитимную разработку, запуская игры от Windows под Linux?



Развивать свое, родное

На самом деле, компенсировать надо отсутствие собственных оригинальных разработок игр на платформе Linux. Introversion Software, независимый разработчик игр, таких как *Defcon*, *Darwinia* и *Uplink*, один из тех немногих, которые активно поддерживают Linux. И это прямое следствие того факта, что разработчики выбирают OpenGL, а не DirectX. Однако Introversion является исключением. Разработчиков, выпускающих версии своих игр под Linux, очень мало – но, возможно, это симптом совершенно другого недуга: разработка игр стоит дорого, а значит, приходится поддерживать ведущие платформы, что, в свою очередь, ведет к использованию DirectX или промежуточных решений.

Отрасль заинтересована не в рыночных нишах, а в играх, эквивалентных развлекательным комплексам, где лучшие голливудские хиты удобно соседствуют с прибыльными точками быстрого питания. Изменение этого соотношения будет более сложным и может быть еще одним хорошим поводом для покупки консоли на проприетарной платформе.

Наигрывая мелодию

Даже такая простая вещь, как воспроизведение музыки, выражает потребность в свободе.

Если вы не играете в игры, то, возможно, слушаете музыку. Но, по крайней мере, купив аудио компакт-диск в магазине или через Интернет, вы почти наверняка сможете прокрутить его под Linux. Существуют некоторые исключения, но гнусная практика печати дисков, не удовлетворяющих требованиям Красной книги, к счастью, отмирает.

Если вы затеяли сделать копию музыкального диска и сохранить ее на своем жестком диске, это уже не так просто. Законность этих действий, особенно в Великобритании, неясна; хотя, будем надеяться, закон скоро изменится. Кое-кто советует покупать и скачивать музыку из интернет-магазинов. Но музыка – это та область, где доминирует злоупотребление цифровыми правами (digital rights mismanagement, DRM). И, понятно, никак нельзя сказать, что DRM и ПО с открытым исходным кодом – задушевные друзья. Трудно предсказать, выберет ли каждый розничный онлайн-торговец музыкальной продукцией свой собственный DRM-механизм, чтобы отличиться от конкурентов, или предпочтет одну операционную систему другой. Скорее всего, это будет нечто среднее. Иногда кажется, что в скором будущем владельцы прав на музыку начнут брать деньги за то, что мы напеваем ее про себя. Филип К. Дик, в этом смысле, должен чувствовать себя как дома.

В лад

Факт тот, что очень немногие онлайн-магазины открыто поддерживают Linux и открытые кодеки, позволяющие скачивать их музыку без помощи их собственного клиента или версии *Windows Explorer*. Независимые продавцы помельче часто предлагают музыкальные файлы в MP3-формате, которые, по крайней мере, без проблем проигрываются под Linux. Без DRM-ограничений можно, например, скачать каталог Warp или новые релизы с Bleep.com. Неплохи и более общие музыкальные магазины, хотя уровень интеграции здесь не на столь высок. Но, как ни странно, предпринимаются попытки это изменить.

Для пользователей Linux лучший пример работы музыкального онлайн-магазина – MagnaTune (http://magnatune.com). Он не только предлагает хорошие условия исполнителям (неэксклюзивные соглашения и разделение прибыли 50/50), но вы можете прослушать музыку до покупки альбома и выбрать между форматами Ogg Vorbis, Flac и MP3, по лицензии Creative Commons. Кроме того, тут нет DRM: вас даже поощряют поделиться музыкой с тремя друзьями. Но лучшее в MagnaTune – поддержка этого виртуального магазина, встроенная прямо в наши любимые музыкальные проигрыватели: *Amarok* в KDE и *Rhythmbox* в Gnome. Оба они обладают возможностью удаленного доступа к коллекции MagnaTune, как будто вся эта музыка находится в локальном каталоге вашего компьютера. Просто перетащите в плей-лист заинтересовавшие вас альбомы и песни, нажмите “Play” – и зазвучит музыка высокого качества, не требуя с вас ни гроша. Если она вам понравится настолько, чтобы ее купить, для этого опять же достаточно просто нажать кнопку. Выглядит революционно, а MagnaTune заслуживает успеха за такой открытый подход к бизнесу.

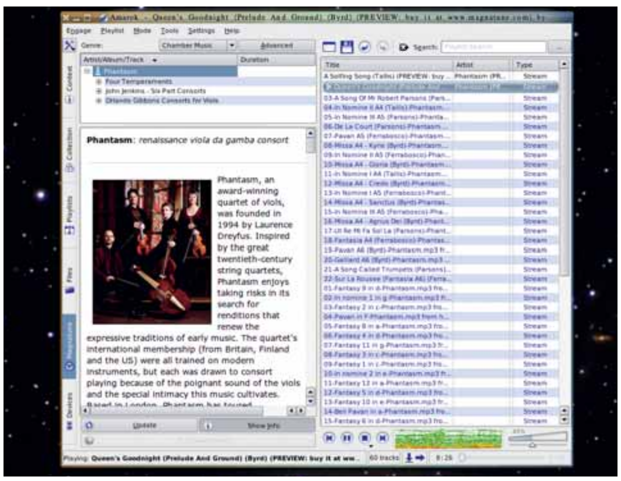
Трудный выбор

Выбор MP3-плейера тоже ложится бременем на совесть пользователей Linux. Подобно эволюционной тактике флоры и фауны тропиков, более всего нас способны прельстить музыкальные проигрыватели, перегруженные функциями, несовместимыми с идеалом открытости. Лучший пример – iPod, созданный для работы только с программой iTunes от Apple: лакомый кусок для покупателей, с прекрасными функциями управления данными и DRM под Windows или Macintosh. Пользователи Linux не у дел, если только не сдадутся на *Wine* или Windows. То же самое – с другими подобными плейерами, от Creative и Microsoft. Если вам нужна гарантированная совместимость с Linux, то найдется разве что незатейливое устройство хранения данных, выполненное отнюдь не в передовой эстетике. Если вы соблазнились штучкой на букву “i”, то получите устройство, совместимое со многими приложениями Linux; но будет ли это правильным шагом? Должны ли мы способствовать разработке устройств, не поддерживающих Open Source, только потому, что они хорошо выглядят, и это примиряет с их проприетарностью? Та же старая проблема.

К счастью, разработчики свободного ПО нашли решение: открытую замену для прошивки многих популярных MP3-плейеров, включая iPod. Она называется *RockBox*, и в большинстве случаев придает добавочные функции, включая поддержку хранения и проигрывания открытых форматов, таких как Ogg Vorbis и Flac (оба отсутствуют в продукции от Apple), и десятки небольших модулей расширения. Имеется также необходимый интерфейс с возможностью выбора дизайна для дисплея. С *RockBox* вам уже не придется принудительно скачивать обновления или беспокоиться о попадании персональной информации в музыкальные файлы.



› Открытый код *RockBox* дал страстным любителям модных примочек право купить сверкающее устройство на букву “i”.



› Интеграция MagnaTune в *Amarok* – самое лучшее, что могут получить пользователи Linux в качестве встроенного музыкального онлайн-магазина.

Ваше мнение

Напишите нам, что вы думаете по этому вопросу, на адрес letters@linuxformat.ru с темой «Какова цена свободы?». Как далеко заходит ваша вера в приложения с открытым кодом и насколько это влияет на жизнь людей вокруг вас? Мы будем рады услышать ваше мнение, так как считаем эту тему важной для обсуждения. Лучшие письма будут напечатаны в одном из следующих номеров.

Разработка программ

Разработчики тоже не могут придти к единому решению, что стоит сделать свободным, а что – нет.

Все, о чем речь шла до сих пор, освещалось со стороны покупателя. Но ведь между разработчиками разногласий быть не может? Увы, и может быть, и есть. Крупнейшие разногласия до сих пор окружают использование коммерческого и проприетарного ПО в крупнейшем проекте с открытым кодом – ядре Linux. Вплоть до середины 2005 года контроль версий для разработчиков ядра Linux осуществлял несвободный *BitKeeper* — хотя тут немалый вклад внес сам Линус Торвальдс, считавший, что, несмотря на свою проприетарность, *BitKeeper* дает столь сильные преимущества для разработки, что их нельзя игнорировать.

Торвальдс открыто критиковал *CVS* и *Subversion*, две наиболее популярных альтернативы, а решение он видел в *BitKeeper*. К сожалению, многие другие разработчики ядра думали иначе. Хотя им предоставлялась бесплатная версия, контроль и развитие приложения по-прежнему держал в руках BitMover, владелец *BitKeeper*. Предложение бесплатной версии было снято в апреле 2005 года, когда основного разработчика ядра, Эндрю Триджелла [Andrew Tridgell], публично обвинили в попытке инженерного анализа частей *BitKeeper* (обвинение он решительно отрицает), и BitMover даже отказался продавать лицензии организации OSDL, где работали Триджелл и Торвальдс, так что продолжать разработку ядра с *BitKeeper* стало неприемлемым.

Именно о подобной ситуации предостерегали критики *BitKeeper* с самого начала: запретительный контроль над продуктом со стороны владельца. Последовали обильные и желчные дебаты через списки рассылки для разработчиков, но о выборе сменщика для *BitKeeper* так и не договорились. В конце концов Торвальдс пообещал и создал совершенно новую систему контроля версий, отличную от *BitKeeper* и *CVS / Subversion*, и назвал ее *Git*. С тех пор *Git* и используется, превратившись в достойный продукт (см. наше *Сравнение* в LXF91).



› **Мигель де Икаса ведет блог о деятельности команды разработчиков Moonlight на http://tirania.org/blog.**

. NET-работы

Но дискуссии, связанные с разработкой ПО, на *Git* не кончились. Другой проект, Mono, тоже расщепил ряды сторонников Open Source. Mono – открытый аналог исключительно влиятельной среды разработки .NET от Microsoft. Если вы с недавнего времени следили за деятельностью разработчиков, то сами знаете, что .NET стала своеобразным стандартом для разработки приложений. И Mono означает не только совместимость с .NET для приложений с открытым кодом, но и практический опыт для разработчиков Linux в одной из самых важных технологий, разработанных за последние пять лет. Но тут снова возникает все та же проблема. Если бы разработчики Linux с самого начала шли путем инноваций, не было бы нужды адаптироваться к решениям собственников. Это главное, за что критикуют Mono, помимо букета нарушений патентных прав. Если наступит момент, когда открытая версия Mono де-факто исчезнет, получится, что все его совершенство загонит разработчиков к .NET от Microsoft, а не к подлинно открытой альтернативе.

Да и на Mono дело не заканчивается. Те же разработчики Mono работают и над *Silverlight* от Microsoft – это комбинация NET и XAML (Extensible Application Markup Language). *Silverlight* является попыткой Microsoft конкурировать с аналогами Flash-технологии от Adobe и JavaFX от Sun для загрузки контента и приложений, встроенных в браузер. Первую рабочую версию *Silverlight* в результате недельного марафона создали Мигель де Икаса [Miguel de Icaza] и его команда, работая по двенадцать часов в день. Планируется, что их проект, прозванный *Moonlight*, будет отслеживать график релизов *Silverlight* от Microsoft и обеспечит совместимость Linux с одним из самых ожидаемых проектов года. Понятно, что *Moonlight* столкнулся с той же критикой, что и Mono (кстати, он использует Mono для компонентов .NET), но, бесспорно, пользователи настольных компьютеров получат большие преимущества, поскольку все мы бьемся с проблемами совместимости форматов. Так не наплевать ли им на политику, которой окружена разработка проектов? LXF

Свобода как опция

Люди решают использовать свободное ПО по различным причинам, и эти причины могут оказывать сильное влияние на принятие других решений. Например, немало людей просто используют свободное ПО, поскольку оно наилучшим образом соответствует их потребностям – тут нет никаких политических мотивов. Свобода – это опция, и каждый из нас взвешивает плюсы и минусы использования открытого ПО, которое эквивалентно коммерческим приложениям.

В случае моего *MythTV*, хлопоты по поддержанию работоспособности (почти круглосуточные), в сочетании с затратами на модернизацию аппаратных средств, сделали его менее полезным для меня, чем коммерческие, закрытые, монополистические и запатентованные решения. Мне было неуютно идти на этот компромисс, но получаемые преимущества победили. С этим сталкиваются все пользователи свободного ПО, в большей или меньшей степени. Я считаю, что успех приложений с открытым исходным кодом и GNU/Linux обязан тому, что мы действуем во «враждебной» среде. Проприетарные разработки ПО, насильственные модернизации, закрытое оборудование, дата-майнинг и угроза нарушения патентов – все это стимулирует разработчиков на создание альтернатив. Неважно, на сколько процентов мы привержены к свободному ПО, коли мы вообще его используем, в той или иной форме: каждый новый пользователь дает и новый стимул. Но это не повод, чтобы бежать и покупать XBox 360.

Наши у руля

Излишне напоминать, какую роль играет IBM в развитии и поддержке Linux. По заданию редакции Linux Format Виктор Федосеев встретился с **Инной Кузнецовой** – человеком, который непосредственно отвечает за то, каким будет это влияние в глобальном масштабе.



Наше Интервью

Инна Кузнецова
Директор по стратегии и развитию бизнеса IBM System p. С мая 2007 года – директор по Linux-стратегии корпорации IBM. Выпускница Московского Государственного Университета, магистр математики и компьютерных наук, имеет степень MBA от Columbia Business School (США).

LXF: Инна, ваша должность звучит как «Директор IBM по Linux-стратегии». Что подразумевается под этими словами и что входит в спектр решаемых вами задач?

ИК: Я отвечаю за стратегию Linux в корпорации IBM. Linux сейчас настолько тесно интегрирован в продуктовые линейки IBM, что фактически в каждом подразделении, в каждой стране у нас есть люди, вовлеченные либо в разработку, либо в продажу, либо в маркетинговую поддержку в области Linux, и их усилия необходимо координировать. Необходима единая стратегия, система управления и контроля над стратегией, нам постоянно нужно отвечать на вопрос: достигаем ли мы целей этой стратегией, как развивается наше партнерство, как продвигается наше участие в различных проектах.

Я руковожу командой, которая отвечает не только за разработку этой стратегии, но и за управление всей системой. К примеру, раз в квартал по результатам своей деятельности я отчитываюсь перед двумя старшими вице-президентами IBM: рассказываю, как обстоят дела с Linux в нашей компании. Естественно, что я в свою очередь собираю руководителей подразделений, которые рассказывают мне о том, как у них идут дела, связанные с Linux. Помимо роли стратега и руководителя, в мои задачи также входит сплочение Linux-сообщества IBM. Ведь Linux в IBM сегодня занимаются сотни и сотни людей. Необходимо, чтобы они регулярно получали информацию о том, что происходит в компании в целом, обо всех изменениях, которые происходят на рынке, чтобы они чувствовали себя частью целого и не потеряли дух Linux.

LXF: В течение продолжительного времени IBM придерживалась подхода, который можно описать словами «Linux – на серверах, Windows – на рабочем столе». С выпуском Lotus Workplace для Linux ситуация изменилась?

ИК: Linux имеет большое будущее не только на серверах, но и на рабочих столах. Linux – молодая система, и она очень быстро развивается. Несколько лет назад настольный Linux был мало распространен. Причины этому – и необходимость высокой технической подготовки людей, использующих данную ОС, и банальная проблема отсутствия программных продуктов, необходимых обычному пользователю каждый день – офиса, графических редакторов и так далее. И, кроме того, во многих организациях есть пользователи, которым приходится в силу тех или иных причин использовать программы, работающие под управлением другой ОС. Переводить такую организацию на Linux можно, лишь обеспечив обмен данными и информацией внутри гетерогенной сети.

С одной стороны, в последнее время Linux как система стал проще и дружественней пользователю, для работы в нем уже не нужны какие-то специальные знания. Есть очень дружественные дистрибутивы – от Novell, Ubuntu, Mandriva, которые очень просто ставить, настраивать и обслуживать. С другой стороны, появились продукты, которые решают проблему обмена данными внутри гетерогенных сетей.

К примеру, месяц назад мы объявили о выпуске продукта Open Collaboration Client, который пока работает только под управлением SUSE Linux Enterprise Desktop и представляет собой продукт на основе технологий Lotus, обеспечивающий единую рабочую среду – почта, обмен сообщениями, графический и текстовый редакторы и т.д. Пользователь, по самым консервативным оценкам, экономит 300–500 долларов на одной лицензии в сравнении с продуктами Microsoft.

Но главное даже не это. Если заказчик хочет устроить у себя гетерогенную среду – к примеру, сети на основе Linux и Windows – то он может это сделать. При этом он унифицирует офисные программные продукты, что позволяет без проблем и недоразумений обмениваться

информацией и данными между пользователями с различными ОС. При этом в тех случаях, когда ему важны критические для бизнеса приложения, работающие под отличной от Linux операционной системой, он может это себе позволить. Мне кажется, что такая унификация и станет ключевым моментом в продвижении Linux на рабочих столах.

LXF: Недавно компания IBM объявила еще об одной Linux-инициативе под названием Big Green Linux. Вы не можете прокомментировать ее суть?

ИК: В августе этого года, компания IBM объявила об инициативе Big Green Linux, в рамках которой она будет помогать своим клиентам использовать Linux для уменьшения энергозатрат вычислительных центров путем консолидации серверов, более эффективного управления ресурсами и балансировки нагрузки. Консолидация серверов на основе Linux позволяет провести как существенную экономию электроэнергии, так и снижение общих затрат на охлаждение офисных помещений, а также обслуживание и администрирование серверов. IBM начала применять этот подход к собственной инфраструктуре. Месяц назад мы объявили о консолидации 3900 наших серверов на 30 мейнфреймов под управлением ОС Linux. По нашим расчетам, эта консолидация только в плане экономии электроэнергии позволит снизить затраты на 80% от ее изначальной величины. За пять лет, таким образом, мы сэкономим достаточно энергии, чтобы удовлетворить годовые потребности небольшого города.

Есть и другие аспекты. Сегодня подлежащие консолидации сервера занимают по площади около 139 футбольных полей. Консолидация приведет к огромной экономии на аренде площадей. Благодаря такой консолидации у нас освобождается огромное количество сотрудников, задействованных на администрировании этих серверов. Мы не планируем увольнений – наш план заключается в переобучении их работе с Linux-системам, чтобы помочь нашим клиентам, планирующим подобные проекты консолидации.

Еще один интересный момент такой консолидации – это вопрос о том, что же нам делать с освобождающимися серверами? Часть из них, относительно новых, мы очищаем от информации, приводим в товарный вид и продаем на вторичном рынке. Часть, которые являются уже относительно старыми, мы утилизируем. У нас есть сервис по утилизации старых серверов, IBM Global Asset Recovery Services, который мы предлагаем и своим клиентам. Так, в прошлом году мы утилизировали оборудование весом более ста миллионов фунтов. И только менее 1% ушло в отходы – т.е. на свалки. Так что этот наш бизнес является довольно экологичным.

Клиенты IBM тоже прибегают к консолидации. Приведу два примера компаний, объявивших в августе этого года о консолидации на серверах системы System p. Первая из них – компания Telefonica Mobiles, третья по размеру телефонная компания в мире и самая крупная телекоммуникационная компания в Испании. Она консолидировала несколько сот своих серверов платформы x86 на двух серверах платформы System p, работающей на базе процессоров POWER. Некоторые приложения были переписаны для Linux на платформе System p, а некоторые используют технологию pAVE (System p Application Virtual Environment), которая позволяет запускать приложения для Linux на платформе x86 под Linux на System p без изменений и перекомпиляции. Данная технология доступна бесплатно для скачивания с нашего web-сайта. Сегодня она еще находится в бета-версии, но специалисты из Telefonica Mobiles были настолько восхищены результатами и стабильностью работы этой технологии, что приняли решение применить ее в постоянной деятельности.

Другая компания, объявившая о проекте консолидации – Volkswagen AG. Она консолидировала 76 своих серверов на 6 серверов опять же System p.

Кроме того, в рамках инициативы Big Green Linux компания IBM объявила новый продукт – Information Server Blade, предназначенный для виртуализации данных. Работает он под управлением Red Hat Linux. IBM Information Server Blade – это полностью интегрированное решение, включающее blade-системы IBM, программную платформу для интеграции данных IBM Information Server и услуги по внедрению, включая финансирование. Новое решение обеспечивает консолидацию и перемещение значительных объемов данных для более глубокого понимания бизнеса и решения проблем, связанных со стремительным ростом объемов данных.

Big Green Linux – это долгосрочная инициатива, сейчас мы находимся в самом ее начале, но будем регулярно информировать прессу и заказчиков о последующих этапах.

LXF: Позвольте спросить, как вы относитесь к альянсу одного из ваших ключевых Linux-партнеров, компании Novell, с Microsoft? Ведь вы в последнее время очень интенсивно работаете с Novell: в частности, в области Big Green Linux и уже упоминавшегося выше Open Collaboration Client. Не повредит ли такое поведение Novell бизнесу и имиджу компании IBM в мире Linux? И как вы считаете, необходимо ли заключение патентных соглашений с компанией Microsoft?

ИК: Big Green Linux существует на всех дистрибутивах, с которыми мы работаем. Наша стратегия – работать с несколькими дистрибьюторами. Мы считаем, что у заказчика должно быть право выбора – это стабилизирует рынок и позволяет технологиям расти быстрее. Так что мы не выделяем кого-то одного, и наши инициативы доступны всем, кто в них заинтересован. В настоящий момент у нас существуют два стратегических альянса – с компанией Novell и компанией Red Hat. Многие наши продукты работают на обоих дистрибутивах. В некоторых случаях какие-то продукты работают только на одном из них – как правило, потому, что второй не заинтересован в этом проекте и просто хочет вложить свои ресурсы в какую-нибудь другую инициативу.

По поводу патентной защиты – я не считаю, что она необходима. Более того, IBM поддерживает Open Invention Network – это организация, предоставляющая юридическую поддержку всем пользователям Linux. Мы считаем, что такая защита адекватна и достаточна, и нет необходимости прибегать к каким-то дополнительным мерам по защите Linux. Несмотря на заявление о нарушении 238 патентов, компания Microsoft до сих пор не назвала ни одного конкретного. Однако, несмотря на расхождение с Novell по этому вопросу, наше партнерство крепко, как никогда ранее. IBM объявила месяц назад, о том что Web Application Server (WAS) Community Edition будет частью SUSE Linux Enterprise Server. Кроме того, IBM и Novell предлагают услуги пользователям, которые хотят мигрировать с сервера приложений JBoss (принадлежащего Red Hat) на WAS. Тем не менее, мы так же работаем с Red Hat, и JBoss работает на наших продуктах.

Однако у этого партнерства есть одно положительного качество. И это – возможность добиться лучшего взаимодействия между Linux и Windows. Оставим в стороне юридическую часть вопроса и сфокусируемся на технических аспектах. Если это партнерство позволит Novell улучшить взаимодействие внутри гетерогенной среды, где есть Windows; если продуктам на основе Linux будет лучше и проще взаимодействовать с продуктами и сетями Windows, то в конечном счете, Linux выиграет. По-моему, это положительный момент такого сотрудничества.

LXF: Давайте вернемся к Open Collaboration Client, который сейчас работает только на дистрибутиве Novell. Скажите, а планируете ли вы выпуск этого продукта для дистрибутивов, отличных от Novell?

ИК: Мы готовы распространить продукт на другие дистрибутивы, под-

Open Invention Network

Open Invention Network – союз компаний IBM, Novell, Philips, Sony и Red Hat, имеющий своей целью содействие в рыночном продвижении операционной системы Linux, покупку патентов и предоставление их на бесплатной основе любой компании, учреждению или частному лицу на том условии, что они не будут использоваться в разработках, конкурирующих с операционной системой Linux или определенными Linux-приложениями.

Open Collaboration Client

Open Collaboration Client – базирующийся на *Eclipse* продукт, включающий в себя бизнес-коммуникации на основе *Lotus Notes*, систему обмена мгновенными сообщениями и совместной работы *Lotus Sametime*, текстовый процессор, электронную таблицу и средство создания презентаций. Имеется возможность добавить и другие компоненты из портфолио Lotus для связи с *Quickr*, *Connections*, *Websphere Portal* и т.д. Слово Open не стоит понимать как Open Source: имеются в виду открытые стандарты.

держиваемые IBM. Корпорация постоянно работает с различными производителями. Если кто-то захочет выпустить данный продукт, то мы готовы к сотрудничеству. Но пока он работает только под Novell, т.к. Novell – первая компания, которая заинтересовалось работой над этим продуктом. В данном случае все определяется вопросами бизнеса. Мы объявили данный продукт всего месяц назад, и фактически мы наблюдаем, есть ли на него спрос. Если спрос появится, то будет разумно выпустить его и для других дистрибутивов или вообще сделать платформонезависимый вариант, как *OpenOffice.org*.

LXF: Недавно прошло голосование по поводу принятия стандарта документов от Microsoft — OOXML — в качестве международного стандарта ISO. Что IBM думает по этому поводу и как IBM относится к стандартам вообще?

ИК: IBM является членом OASIS. Мы считаем, что стандарты должны быть открытыми и не завязанными на одну компанию.

Как вы уже знаете, 4 сентября стандарт OOXML был отклонен. Согласно правилам ISO, у Microsoft есть возможность в следующем году заново поставить этот вопрос на голосование, исправив те замечания, что были высказаны. С моей личной точки зрения, очень важно, что наиболее быстро развивающиеся страны – Китай, Бразилия, Индия – поддержали OpenDocument и не поддержали OOXML. Там, где происходит быстрый рост, очень четко понимают, что нужно для эффективного ведения бизнеса, понимают, что нужно не усложнять процесс, а использовать открытые стандарты.

Мы активно проводим поддержку OpenDocument как член альянса OASIS. В этом, как и многих других видах деятельности, связанных с Linux и открытыми стандартами, IBM является активным членом сообщества. Так, мы поддерживаем сообщество Linux за счет наших патентов и разработок – например, свыше 600 разработчиков IBM в 40 странах заняты в нашем Центре Технологий Linux (Linux Technology Center). Из них более 300 заняты разработками исключительно в сфере ядра Linux. И специалисты этого центра очень активно участвуют в работе над открытыми стандартами.

Мы подаем любые изменения кода в сообщество Linux в строгом соответствии с существующим процессом, поддержка сообщества – это залог честной конкуренции в общей борьбе за лучший код и его максимальную эффективность. Так же и в России – мы работаем совместно с Linux-сообществом, развивая его и помогая ему, но ни в коем случае не конкурируя и не выделяясь из него.

LXF: Каковы перспективы развития собственных операционных систем IBM (AIX и z/OS) в свете развития Linux?

ИК: Linux и ОС IBM будут развиваться параллельно. AIX разрабатывается специально под аппаратное обеспечение IBM System p, а z/OS – под IBM System z. Linux разрабатывается огромным сообществом, и он всегда хуже оптимизирован под конкретную платформу, чем операционная система того же производителя, который поставляет аппаратные средства – и лучше оптимизирован в целом, независимо от аппаратной платформы.

Одним заказчикам важнее оптимизация под разные платформы, другим – те преимущества, которые они могут получить от ОС, оптимизированной под конкретные аппаратные средства. Одни выберут Linux, другие – AIX или zOS на соответствующей платформе. Безусловно, мы будем интенсивно развивать наши собственные ОС. В прошлом году мы отметили двадцатилетие ОС AIX, а в этом году выпускаем версию AIX 6.0 с множеством новых возможностей.

LXF: Вы — представитель крупной компании. Было бы интересно узнать ваши взгляды на так называемую «коммерциализацию Linux», которую иногда ставят ему в укор любители BSD.

ИК: Наличие выбора всегда прекрасно. Наличие Linux не служит ограничением на развитие других инициатив. Но у коммерциализации существует два совершенно разных аспекта – это вопрос разработки и вопрос внедрения. Если коснуться первого из них, то наличие сообщества очень важно. Это сильная сторона Linux. Ни одна компания не может себе позволить нанять 60 тысяч разработчиков и выпускать несколько версий ядра в год. А ведь именно благодаря этому Linux быстро развивается, в нем оперативно и эффективно решают вопросы безопасности, и т.д. Попытки перехватить инициативу в разработках – это ошибка. Компания может при этом выиграть битву, но проиграть войну, так как в долгосрочной перспективе это крайне плохо. И потому-то, несмотря на огромные инвестиции ресурсов в Linux, мы работаем в этом плане именно с сообществом. Если какое-либо из наших предложений не пройдет в финальный код, это может выглядеть как проигрыш небольшого сражения, но в долгосрочной перспективе это означает, что сообщество выберет более сильный вариант, что важно для всех, в том числе и для IBM.

Другой вопрос – внедрение. Linux вышел из периода евангелизма – он вырос, это следующий этап эволюции. Это система, на которой уже сегодня работают критические для бизнеса приложения. Так, к примеру, одна крупная компания в США по производству моторов отслеживает всю цепочку их производства – от цеха до продажи при помощи сенсоров, меток RFID и систем на основе ОС Linux. Linux применяется там, куда вкладываются немалые деньги. И для качественного, профессионального обслуживания, внедрения и обучения необходимы серьезные коммерческие компании. Ведь в конечном итоге это ведет к тому, что деньги уходят и к программистам тоже: должны же они получать зарплату. Чем больше мы привлекаем ресурсов, тем лучшими программистскими кадрами можем оперировать.

Такая коммерциализация – это естественный процесс, это рост, это эволюция, и это нормально. Если продукт не растет, то начинается стагнация и последующее его умирание.

LXF: Каким, по-вашему, будет «мир Linux» через пять лет? Займет ли, на ваш взгляд, Linux в течение этого срока заметную долю на рабочих столах?

ИК: Я думаю, что мы, несомненно, увидим огромное продвижение Linux в области рабочих столов. Помимо появления новых продуктов с необходимой функциональностью, позволяющих компаниям сэкономить деньги, мы также увидим, как Linux идет в школы, ВУЗы и государственные учреждения. Очень много стран в мире, включая Россию, начинают тестировать, а часто и внедрять продукты на основе Linux в государственных и учебных заведениях. Более того, к примеру, в США этот процесс давно идет. Во многих университетах, во многих колледжах Америки Linux уже давно используется. Моя дочь, к примеру, учится в MIT, и на ее настольном компьютере стоит Linux. В данном ВУЗе Linux – очень распространенная система.

Факт прихода нового поколения в Linux очень важен. Когда придет целое поколение, которое знает Linux и сможет сделать выбор в более информированном пространстве, это безусловно пойдет на пользу всем.

LXF: А вы сами используете Linux у себя дома? Если да — то какой?

ИК: Да, я и моя семья, мы используем Linux у себя дома. У нас несколько компьютеров, и мы иногда тестируем новые дистрибутивы. Можно провести ревизию своих домашних компьютеров, но я уверена, что там найдутся несколько популярных Linux'ов. LXF



Как сделать Slashdot'о-упорным ваш сервер

Ваш web-сервер подорвал силы, отбивая наплыв посетителей – аж по два обращения в секунду? **Пол Хадсон** покажет, как его излечить.

Digg, Reddit, Delicious, Furl, Newsvine и другие крупные сайты новостей посещаются миллионами пользователей, но, вообще говоря, лишь один из них популярен достаточно, чтобы дать имя явлению, вгоняющему в пот сисадминов: Slashdot-эффекту. Новости на нем, возможно, вам и не по вкусу, но Slashdot остается едва ли не главной отрадой хакеров в Интернете. Ссылка на ваш web-сайт, включенная в одну из новостей с главной страницы, приведет к вам за несколько часов более сорока тысяч человек, что чревато двумя исходами: либо огромным ростом прибыли от рекламы (больше народу прокликает ваши баннеры), либо тем, что ваш сервер расплавится, погрузив сайт в анабиоз ожидать волшебного поцелуя, который вернет его к жизни. Более того, если ваша история интересна широкой аудитории, то ее переопубликуют сотни других сайтов, источником новостей для которых является Slashdot, и посетителей может быть в несколько тысяч раз больше – если, конечно, ваш сайт не рухнет.

Предупреждение: до старта

Внесение изменений в настройки вашего сервера может привести к непресказуемым последствиям. Мы советуем использовать для этих тестов не занятый в основной деятельности сервер и перед внесением любых изменений в настройки делать резервную копию всех файлов системы. После изменения настроек Apache/MySQL не забудьте перезапустить сервисы, чтобы новые параметры вступили в силу.

Но вот беда: большинство сайтов именно что рухнет. Slashdot-эффект получил свое название из-за того, что орды нагнанных им визитеров превышали пределы возможностей сайтов, и последние в конце концов падали. Если же вы настроены серьезно и хотите, чтобы ваш сайт был доступен всегда и везде, то есть много способов сдержать напор посетителей. Да, надо взяться за оружие и показать прыщавым слэшдоттерам, что вам плевать на 500 запросов в секунду, что вам приятно быть как Dugg и что Delicious остается только отирать с лица грязь, летящую из-под ваших копыт. Короче, пора превратить ваш медленный и средненький web-сервер в супербыстрый и супернадежный сервер вашей мечты.

Живи и дай умереть

[В заголовке – название известной песни Пола Маккартни, – *прим. пер.*]

Быстрота вашего кода особого значения не имеет: в конечном итоге все зависит от грамотности настройки *Apache*. Проблема в том, что люди думают: «Ну вот, сайт протестирован: он справляется, даже когда на него заходит сразу 20 человек – отлично!» 20 обращений в секунду – это неплохо, если вы размазываете число посещений в месяц равномерно по месяцу. Когда мы спросили, насколько опасен Slashdot, у эксперта **LXF** по web-технологиям Майка Сондерса, тот побледнел, задрожал и не сразу ответил: «Slashdot-эффект – это как тысяча термитов, грызущих ваши сетевые кабели». И когда он говорил о тысяче, он и имел в виду тысячу: двадцать обращений в секунду – это неплохо, но если ссылка на ваш сайт угодит на Slashdot, реально получить до тысячи обращений в секунду.

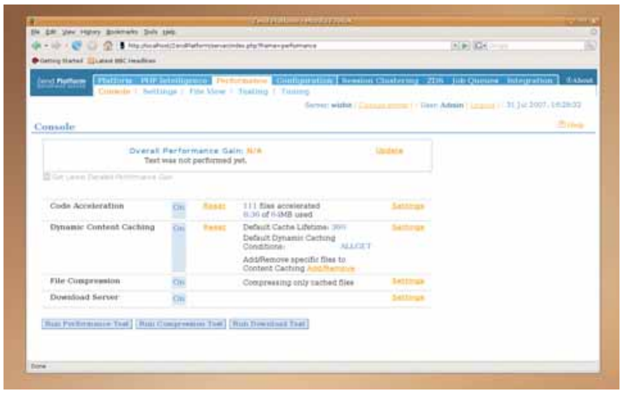
Первый эффект Slashdot возникает тогда, когда из-за огромного количества одновременно открытых соединений посетителям не удается соединиться с сервером, даже если он еще жив и нормально обрабатывает запросы. И они нажимают на кнопку «Обновить», и делают это снова и снова; и рано или поздно сервер в самом деле падает. Вот вам и второй эффект Slashdot: ваш сервер становится дымящейся развалиной.

Таким образом, решение проблемы состоит в выборе: обеспечить пользователям либо 100%-ную производительность ресурса при низкой загрузке и 5%-ную при высокой, либо 90%-ную при низкой и 70%-ную при высокой. Большинство людей предпочло бы второй вариант, потому что большую часть времени почувствовать разницу между 90% и 100%-ной производительностью невозможно. Но когда на сайт заходят тысячи посетителей и время имеет значение, различие между этими вариантами становится очевидным.

Эти два варианта разделены одной строкой в файле настроек *Apache*: *KeepAlive*. Она появилась в HTTP/1.1 и предназначена для того, чтобы разрешить постоянное соединение между клиентом и сервером. Браузер без поддержки *KeepAlive* соединяется с сервером, загружает страничку, отсоединяется, затем проверяет, есть ли на странице рисунки и другие вложения, после чего создает отдельные соединения для каждого из этих объектов. Это требует массы дополнительных действий, поэтому *KeepAlive* сохраняет одно соединение открытым около 15 секунд, позволяя пользователям загрузить в его ходе несколько файлов. Если на вашем сайте много графики, то включение поддержки *KeepAlive* способно уменьшить время ожидания (это не то же самое, что время загрузки страницы) примерно вдвое.

Все это хорошо, но вспомним, что *Apache* поддерживает не более 256 одновременных соединений. Так что если пытаются подключиться 1000 пользователей, то *Apache* откроет соединения с двумястами пятьюдесятью шестью из них, обработает их запросы на все страницы и картинки, потом оставит соединение открытым на 15 секунд на случай, если клиенты затребуют еще что-либо, и, наконец, закроет их. Тут можно перейти к обслуживанию других пользователей. Здесь есть две основные проблемы. Во-первых, глупо сохранять соединение открытым в течение 15 секунд, когда его ждут сотни других пользователей. Во-вторых, приоритет загрузки изображений над загрузкой содержимого – штука неправильная по сути. Если отключить поддержку *KeepAlive*, то файл журнала запросов будет выглядеть примерно так:

User 1 запросил index.html
User 2 запросил index.html
User 3 запросил index.html
User 1 запросил foo.jpg



› *Zend Platform* **отслеживает наличие проблем с PHP на web-сервере, а также автоматически ускоряет выполнение скриптов.**

User 2 запросил foo.jpg
User 3 запросил foo.jpg
User 4 запросил index.html
User 1 запросил bar.jpg
User 2 запросил bar.jpg
User 3 запросил bar.jpg
User 4 запросил foo.jpg
User 4 запросил bar.jpg

Это куда справедливее, и вероятность того, что будет обслужены все пользователи, а не несколько приоритетных, намного выше. Убедились? Так и должно было быть! Чтобы отключить поддержку *KeepAlive*, откройте файл настроек *Apache* (обычно **/etc/httpd/httpd.conf** или **/etc/apache2/apache2.conf**) и измените строку

```
KeepAlive On
```

на строку

```
KeepAlive Off
```

После перезапуска *Apache* разницы в скорости вы почти не заметите. Но когда наступит критический момент и ваша история станет первой на Slashdot, тогда вы почувствуете разницу.

Оснастка PHP

Итак, *Apache* заработал на полной скорости; мы можем обратить наш взгляд на PHP, на котором написано большинство популярных web-сайтов. С PHP связаны две основные проблемы:

1 Это интерпретируемый язык, и скрипт должен компилироваться каждый раз, когда запрашивается страница. Как ни странно, время компиляции часто гораздо больше времени выполнения.

2 Код многих популярных проектов оставляет желать лучшего или слишком раздут. Примерами последнего могут служить *PHPBB* и *PostNuke*. Но даже если качество кода очень высокое (например, *MediaWiki* или *Drupal*), первая проблема все равно остается.

Есть и еще одна проблема, но от нее обычно страдают только системные администраторы Windows: если интерпретатор PHP запускается в режиме CGI, то он работает ужасно медленно. В Linux PHP чаще всего уже сконфигурирован для запуска как модуль *Apache*, так как этот способ обеспечивает наилучшее быстродействие. Если вы не уверены, создайте файл **info.php** в корневом каталоге web-сервера и добавьте в него следующие строки:

```
<?php
phpinfo();
?>
```

Загрузив этот файл в браузере, в разделе **Server API** вы должны увидеть строку **Apache 2.0 Handler.** Если ее нет, то PHP работает не с максимальной производительностью, и вам нужно установить пакет *mod-php5* для сервера *Apache*.

Но даже если PHP настроен правильно, ваша работа только начинается: вам предстоит установить кэш и оптимизатор кода, разобраться, как сгенерировать отчет, если страницы загружаются слишком долго; наконец, все это должно работать как можно быстрее, потому что наша цель здесь – одолеть Slashdot, а не тратить кучу времени на редактирование файлов конфигурации.

Моментальное решение – *Zend Platform*, которая умеет все вышепе- »

Управление несколькими серверами Apache

Ваш сервер *Apache* может поддерживать не более 256 одновременных соединений. Это много, но недостаточно для победы над Slashdot'ом, поэтому ваша цель – обработать эти запросы как можно быстрее, чтобы к серверу могли подсоединиться другие пользователи. Однако ваш сервер может быть не настроен на обработку такого количества одновременных соединений – загляните в файл настроек *Apache* сразу после настроек модуля *mpm_prefork* и проверьте значения параметров **StartServers**, **MaxClients** и **MaxRequestsPerChild**. Первый определяет количество серверов, создаваемых при запуске сервера, второй – количество серверов, для которых разрешен запуск, и третий – сколько запросов должен обрабатывать каждый сервер, прежде чем он будет снят. Если нагрузка вашего сайта высока, можно увеличить значение **StartServers** до 32, при этом убедившись, что значение **MaxClients** также высокое.

Однако здесь есть одно условие: сервер *MySQL* может обрабатывать ограниченное количество соединений. И если серверов *Apache* больше, чем соединений *MySQL*, то узкое место системы смещается к *MySQL*, и вам нужно увеличить количество соединений в настройках сервера *MySQL*.



» речисленное и даже больше. Эта среда хороша тем, что с ней можно ни о чем не думать – вы ее устанавливаете, и она автоматически кэширует и ускоряет выполнение всех страниц на сервере без вашего участия. Если сайт достаточно крупный, то вам, возможно, захочется увеличить количество оперативной памяти, выделяемой под кэш – по умолчанию это 64 МБ, вполне достаточно для кэширования тысячи PHP-скриптов приличного размера. Вспомним, что кэш кода позволяет компилировать скрипты PHP лишь один раз, сохраняя в оперативной памяти результат компиляции, что ускоряет их выполнение. В результате ваш сайт работает не хуже, чем раньше, а число обращений в секунду увеличивается в три-четыре раза.

Если вы хотите еще подхлестнуть свой сервер, попробуйте динамическое кэширование содержимого: страниц, не изменяемых обращением к ним. Например, допустим, форма с запросом **GET** вызывает только получение данных, а запрос **POST** преобразовывает страницу. Динамическое кэширование страниц в *Zend Platform* позволяет избежать ненужных обращений к базе данных для тех страниц, содержимое которых не меняется. На практике хранение кэшированных страниц в оперативной памяти обеспечивает дополнительное повышение скорости в 4 раза по сравнению с традиционными системами с кэшированием на жестком диске.

> **«Кэш запросов – одна из самых недооцененных характеристик MySQL.»**

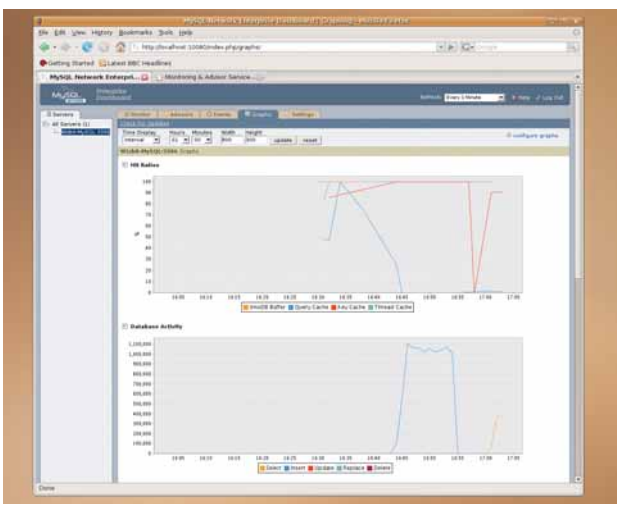
Если вы еще не убеждены, что *Zend Platform* может снять с вас тяжкое бремя, подумайте вот о чем: эта система бесплатна для разработчиков. Это означает, что пока эта платформа не работает на настоящем сервере, вы можете пользоваться ей совершенно безвозмездно. Стоимость лицензии на продукт – 685 фунтов в год без учета НДС, включая поддержку через Интернет.

Мой, мой MySQL

Многие используют для *MySQL* движок *MyISAM* – прежде всего потому, что он работает очень быстро, заставляя забыть о потенциальной возможности повреждения данных. Но если что-то уже с ходу работает быстро, это не значит, что нельзя добиться еще большей скорости. Этим мы сейчас и займемся, поскольку, кое-где подправив файлы конфигурации, можно реально увеличить скорость работы *MySQL*. Правда, с оговоркой: чтобы достичь максимальной производительности, потребуется много оперативной памяти.

Прежде всего откройте файл настроек *MySQL* – обычно это **/etc/my.cnf**, **/etc/mysql/my.cnf** или что-то в этом роде. В вашем дистрибутиве, скорее всего, уже есть файл с настройками по умолчанию, но могут быть и файлы с настройками для различных типов серверов. Запустите **updatedb** для обновления локальной базы данных, затем выполните команду:

```
locate my-
```



› **Сервер *MySQL Enterprise* автоматически выводит информацию о числе обращений к кэшу и числе соединений, а также другую полезную статистику, которая поможет вам настроить сервер.**

Постоянные соединения: друг или враг?

PHP позволяет устанавливать постоянные соединения с сервером *MySQL*. На практике это означает, что каждый поток сервера *Apache* создает свое собственное соединение с базой данных. Преимущество такого подхода в отсутствии создания и закрытия соединений каждый раз, когда выполняется новый скрипт. Но это также означает, что сервер *MySQL* должен поддерживать как минимум столько одновременных соединений, сколько запущено серверов *Apache*. Второе неудобство – многие приложения PHP не поддерживают постоянных соединений; однако в вашем собственном коде с этим легко справиться.

В Ubuntu эта команда находит файлы конфигурации *MySQL* в каталоге **/usr/share/doc/mysql-server-5.0/examples**, например, **my-huge.cnf** с конфигурацией *MySQL* для крупных серверов. Изучая эти файлы, можно узнать много нового, особенно потому, что в начале файла содержатся комментарии, поясняющие, для каких систем данный файл больше всего подойдет.

Если по-простому, учитывать надо следующие параметры:

- Размер ключевого буфера.
- Размер кэша запросов.
- Настройки совместного выполнения потоков.
- Размер кэша таблиц.

Это не полный список кэшей и буферов, используемых *MySQL*, но только изменение именно этих параметров приводит к заметным результатам.

Проверить, достаточен ли объем ключевого буфера, очень просто. Соединитесь с сервером *MySQL* и в командной строке выполните следующую команду:

```
SHOW STATUS LIKE '%key_read%';
```

Вы получите два числа: Key_read_requests и Key_reads. Первое показывает, сколько раз в базе данных производилось считывание ключа индекса, а последний – сколько раз ключ не удавалось найти в кэше, и он считывался с диска. Для большинства сайтов необходимо, чтобы значение Key_read_requests было как можно больше по сравнению с Key_reads. Определить числовой показатель качества можно с помощью простой формулы:

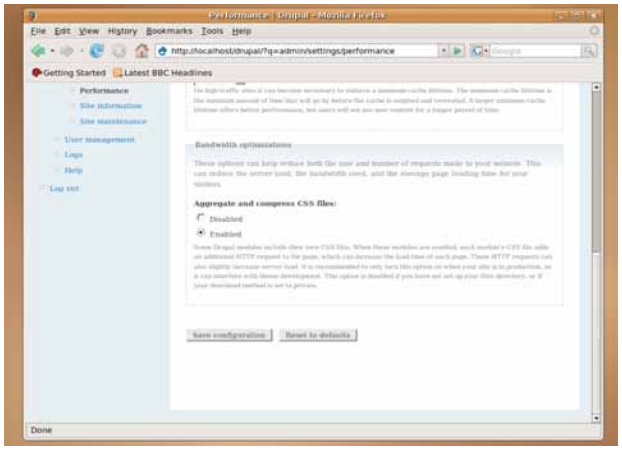
100 – ((Key_reads / Key_read_requests) * 100)

Вы берете значение Key_reads и делите его на Key_read_requests, затем умножаете результат на 100 и вычитаете получившееся число из 100. Готово. Например, если у вас 30 Key_reads **и** 1177 Key_read_requests, нужно разделить 30 на 1177, получится 0.025, затем умножить это число на 100, и получится 2.55, наконец, вычесть это число из 100, и получится 97.45%. Это означает, что 97.45% всех ключей индекса считываются из кэша – это великолепно! Если у вас получилось менее 80%, то либо у вас проблемы, либо операций вставки, обновления и удаления данных просто очень много, что приводит к росту Key_reads. Если результат составил от 80 до 95 процентов, то нужно увеличить размер ключевого буфера – 256 МБ считается более или менее нормальным для больших баз данных. Если у вас получилось более 95%, беспокоиться не о чем.

Кэш запросов – это одна из самых недооцененных характеристик *MySQL*. Он особенно удобен, когда объем данных большой, запросы выполняются часто, а данные изменяются редко. Когда вы выполняете запрос, объем результата которого не превышает 16 МБ, то этот результат помещается в кэш. Если тот же запрос поступает снова, его результат берется из кэша. Фокус состоит в том, что если данные в какой-либо из таблиц, участвующих в кэшированном запросе, изменяются, кэш автоматически очищается, чтобы результат снова формировался непосредственно из базы данных.

В файле настроек *MySQL* размер кэша запросов определяется переменной query_cache_size в разделе [mysqld]. Например, рекомен-





› **Многие приложения PHP содержат собственные средства оптимизации, но внимательно проверяйте их, чтобы убедиться, что вы в выигрыше!**

дованный минимум составляет

```
query_cache_size=16M
```

"M" в *MySQL* означает МБ; таким образом, эта строка устанавливает размер кэша в 16 МБ. Но если размер оперативной памяти позволяет – а он должен – установите это значение в 64M, чтобы кэшировалось больше запросов.

Последние два параметра лучше сделать побольше. Для кэша таблиц обычно вполне достаточно 128; если у вас больше 128 таблиц, просто увеличьте это число. Что касается одновременного выполнения потоков, то число параллельных потоков должно быть вдвое больше числа ядер в вашем процессоре – стало быть, для четырехъядерного процессора установите его в 8. Вот и все: перезапустите сервер *MySQL* для принятия новых настроек и наслаждайтесь обретенной скоростью!

Улучшенный мониторинг MySQL

MySQL входит в большинство дистрибутивов, поэтому многие не знают, что существует превосходная коммерческая версия продукта – *MySQL Enterprise*. Эта версия использует тот же сервер, что и *MySQL*, но с выходящими каждые три месяца пакетами обновления, службой поддержки, и – главное – сетевым монитором *MySQL*. Это очень толковый набор программ, который локально запускает собственный web-сервер и сервер *MySQL*, подключаемые ко всем остальным серверам *MySQL* и формирующие подробный отчет об их работе. Если один из серверов перестает отвечать на запросы, или загрузка его процессора или использование оперативной памяти достигает предельного значения, сетевой монитор отправляет сообщение об этой проблеме на web-интерфейс администратора и даже выводит небольшие графики использования ключевых буферов, числа соединений и т.д.

Кроме того, предоставляется консультативный сервис *MySQL*, который сканирует *MySQL*-сервер и формирует список возможных проблем: нет ли правил, которыми вы пренебрегаете? Возможны ли проблемы с безопасностью в вашей конфигурации? Самым полезным из таких сервисов являются советы по использованию памяти: ведется наблюдение за использованием буферов и даются рекомендации по оптимизации настроек – вся умственная работа выносится за скобки…

Настройки для отдельных приложений

Вместе с *MySQL Enterprise* вам предоставляется доступ к службе поддержки при возникновении каких-либо проблем, однако пока вы не особо налегаете на *MySQL* (особенно с репликацией), проблемы вряд ли возникнут. В любом случае, следует попробовать 30-дневную тестовую версию *MySQL* Enterprise – установить ее на свой сервер, дать ей поработать несколько дней на настоящем сайте и посмотреть, не посоветует ли эта система что-нибудь полезное.

Некоторые сложные приложения на PHP содержат встроенные средства оптимизации, помогающие ускорить их выполнение, но если вы пользуетесь *Zend Performance*, то постарайтесь не вмешиваться в его работу. Например, *MediaWiki* (приложение, на базе которого работает Wikipedia) имеет возможность кэширования страниц (опция memcache). «Скомпилированные» страницы хранятся в оперативной памяти и при необходимости извлекаются оттуда. Это может реально увеличить производительность, так как *MediaWiki* – крупное приложение, очень требовательное к системным ресурсам при преобразовании кода из нотации wiki в HTML.

Но дело в том, что *Zend Platform* способно обеспечить гораздо лучшее кэширование, чем *MediaWiki*, и, объединив кэши, можно использовать оперативную память наилучшим образом. Помните, что memcache – это общий способ кэширования, тогда как *Zend Platform* предлагает решение, направленное на PHP: она кэширует скомпилированные скрипты и выводимые страницы и автоматически отслеживает изменения на сайте, поэтому всегда понятно, когда нужно обновить кэш. В *MediaWiki* вы в лучшем случае получаете кэш вывода, который не идет ни в какое сравнение.

Другие популярные проекты содержат различные средства оптимизации, которые нужно изучить более внимательно. Например, *Drupal* содержит средства оптимизации кэширования и сжатия данных, но мы опять же рекомендовали бы вам оставить кэширование отключенным (как по умолчанию), зато попробовать включить сжатие CSS. До и после внесения изменений всегда снимайте контрольные характеристики, чтобы убедиться в том, что вы случайно не замедлили работу системы.

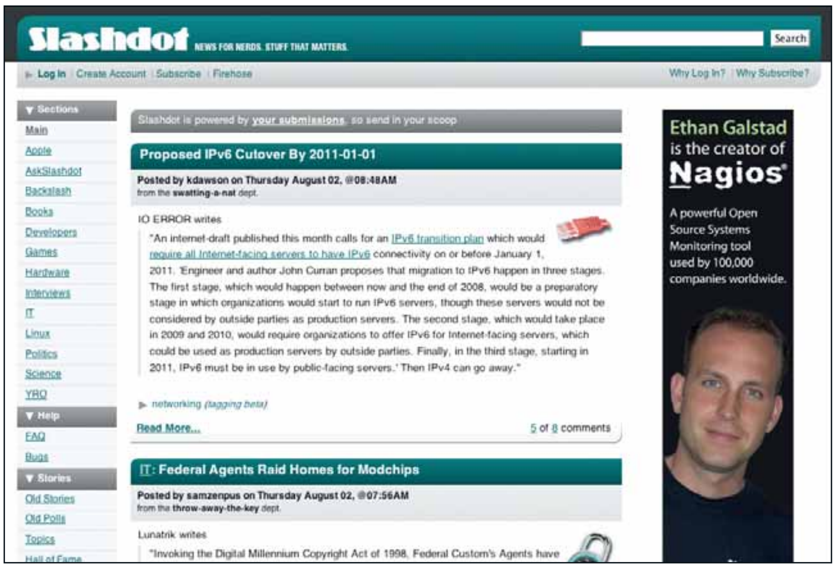
Итак, наше краткое руководство по оптимизации web-сервера подошло к концу. Применив эти настройки, вы сможете пережить испытание не только Slashdot'ом, но и Slashdot'ом в сочетании с Digg, StumbleUpon, Furl, Reddit, Delicious и другими крупными сайтами – на деле вас ограничивает лишь объем оперативной памяти и частота процессора. Вам осталось только придумать что-нибудь этакое, чтобы ваш сайт засветился на Slashdot, а вот с этим, боюсь, я помочь уже не смогу… LXF

Тестирование нагрузки

Прежде чем вносить изменения в конфигурацию *Apache* или *MySQL*, снимите характеристики вашего сайта. Лучший способ это сделать – это запустить на другом компьютере утилиту **ab**: она несколько тысяч раз обращается к серверу и оценивает время отклика. Для начала запустите эту команду

```
ab -n 10000 -c 100 -k http://localhost/index.php
```

Она отправит на ваш сервер 10000 запросов, разделенных на сто одновременных соединений. Ключ **-k** означает «использовать поддержку KeepAlive», что позволит вам оценить характеристики сервера с поддержкой *KeepAlive* и без нее.



Что за штука... OpenMoko



Считаете ли вы изобретателем телефона Шарля Борселя, Филипа Рейса, Антонио Меуччи или Александра Грэма Белла, **Ник Вейч** найдет, что вам возразить.

» Я так вижу, что речь опять пойдет о заморочках с открытым ПО. О чем сегодня? Очередная система управления контентом?

Да, вы правы, это открытый проект, но не совсем тот, что можно запустить на вашем web-сервере. Ну разве что на очень, очень маленьком…

» Смеетесь? А я-то думал, что вы тут все объясняете – так что давайте, делитесь…

OpenMoko – это название проекта, разрабатывающего новую мобильную платформу. Цель проекта – свободная и открытая платформа для разработки приложений к мобильным телефонам, что может привлечь больше производителей мобильных устройств к установке на них Linux.

» Минуточку, но ведь телефонов на Linux и без того полно?

Да не так уж и полно. Motorola выпускает несколько устройств на базе Linux, но они предназначены в основном для японского рынка. Конечно, есть Greenphone от Trolltech, который вышел в августе прошлого года, но что-то не заметно толпы компаний-производителей, жаждущих принять участие. Впрочем, «кривых» устройств хватает.

» В чем же отличие OpenMoko?

Дело не в отличиях, это вообще другая платформа. Но в некотором смысле разработка более открыта: кроме проприетарных драйверов для GPS и GSM, все остальные исходные коды вышли под открытыми лицензиями и могут быть вполне законно модифицированы любым желающим.

Ядро OpenMoko базируется на обычном ядре 2.6.21 и оснащено некоторыми открытыми усовершенство-



ваниями пользовательского интерфейса и доступа к оборудованию, но основная часть не особо завязана на Linux. Например, для подключений по Bluetooth и GSM используются стандартные открытые библиотеки.

А это хорошо?
Судите сами, ведь когда обе платформы – и мобильного устройства, и настольного компьютера – пользуются одинаковыми библиотеками, насколько же упрощается портирование приложений, не говоря уже о разработке и тестировании. По-моему, вскоре следует ожидать обычного наплыва портов *Doom* и игрушек-головоломок. В области взаимодействия с пользователем в настоящее время предлагаются *GTK* и als.

***GTK*? Это чтобы проще было Gnome позвонить?**
Эй, шучу здесь я…

Ну, раз уж вся разработка открыта, теперь пользователи оседлают сотовых операторов и вообще перестанут платить за переговоры, а?
Да нет, вряд ли. В конце концов, соединение происходит через провайдеров, а уж они-то пристально следят, что можно делать в их сетях, а что – нельзя.

Какой же тогда во всем этом толк?
А толк состоит в том, что разрабатывается универсальная, функционально богатая платформа, позволяющая извлечь максимальную пользу из «продвинутой» аппаратуры, и она упростит разработку приложений для мобильных устройств, да и пользователи не будут забыты. Представьте себе будущее, когда, обновляя свой мобильник, вы сохраните свои данные и настройки и просто перенесете их на новую трубку: ведь все приложения будут там работать. Да и просто сможете запускать свои любимые программы на мобильном телефоне.

Здорово. Какие же телефоны поддерживаются, и где можно их купить?
Для конечного пользователя OpenMoko пока не имеет стабильной версии. Хотя целью является платформа, пригодная для установки на любой телефон, в настоящее время разработка сконцентрирована на единственном устройстве, FIC1973. В свое время этот телефон прочили на замену iPhone, хотя, разумеется, iPhone уже с нами. Тем не менее, FIC1973 смотрится неплохо и обладает похожим интерфейсом через сенсорный экран. В наличии GPS и прочие функции, хотя HDSPA, развивающийся стандарт мобильной широкополосной связи, не поддерживается тоже.

Я и не слыхивал о такой фирме. Неужели лидеры проекта не могли найти более известного брэнда?
Ну, брэнд FIC широко известен в Китае. А поскольку основная часть разработчиков работает в FIC и вся поддержка исходит от него же, неудивительно, что за основу команда взяла один из собственных телефонов. Да вы не думайте, что FIC – убогое устройство. Текущая версия оснащена 128 МБ ОЗУ, 64 МБ флэш-носителем и слотом для microSD. Здесь есть и GPS, и GPRS/GSM, и прекрасный VGA-экран. Хотя они и пытаются разработать платформу и добиться ее стабильности на отдельно взятом устройстве, в дальнейшем возможен перенос системы и на другие телефоны.

В самом деле? Что-то этого не заметно…
Возможно, но ведь в OpenMoko участвует не одна команда, в проекте заинтересованы многие разработчики и организации. Один из энтузиастов ухитрился установить систему на свой Palm Treo.

Ну хоть про это я что-то слышал. Так для каких еще телефонов годится новая платформа?
Официально поддерживается пока лишь FIC1973, но видятся перспективы переноса платформы на подобные устройства, оснащенные чипами ARM. На рынке мобильных телефонов процессоры ARM вездесущи, поэтому о возможностях речь не идет, остается лишь руки приложить. Понятно, что в основном усилия OpenMoko будут направлены на телефоны с сенсорными экранами, а это разом вычеркивает многие потенциальные приложения с другими интерфейсами.

> «Представьте себе, что в будущем вы сможете перенести все свои данные, настройки и софт на новый гаджет.»

Черт возьми, я посмотрел спецификацию телефона и обнаружил, что в нем нет даже WiFi. Не проще ли взять две пустые консервные банки, и при помощи натянутой струны…
Покамест FIC1973 – тоже опытный образец. Телефон будет обновлен – вероятно, в этом году – и снабжен новой спецификацией. Есть сведения, что он будет оснащен более быстрым процессором и WiFi на основе *Atheros*.

Хм, а пока что — покупать недоделанную трубку?
Для первого поколения устройств на сайте OpenMoko доступен комплект разработчика. Разумеется, вы будете предупреждены о том, что кроме базовых функций приема и отправки телефонных вызовов, с телефоном мало что можно сделать, если вы сами не возьметесь за разработку ПО. Даже обновленная модель вряд ли будет предназначена для обычных пользователей (ну, разве что для тех, кому хочется казаться крутым).

И за эти «привилегии» предлагается еще и платить?
Телефон первого поколения стоит $300, а $540 – цена с полным комплектом разработчика, в который входят инструменты для перепрошивки телефона и работы с загрузочным образом.

Ну, а если у меня нет лишних $400, чтобы выложить за телефон, который к тому же и не на 100% работает – могу ли я участвовать в OpenMoko?
Конечно. Комплект разработчика, как уже было сказано, создан под FIC1973. Но ведь это всего лишь модифицированная версия Linux для процессора ARM – ее нетрудно эмулировать с помощью Qemu, настроить и т.д. Разумеется, эмулировать некоторые аппаратные функции настоящего телефона невозможно (например, функции GSM/GPS), зато можно разрабатывать и тестировать приложения пространства пользователя или загружать программы, разработанные для устройства другими людьми. Фактически, большинство программ будут основаны на *GTK*, а это значит, что их можно смело портировать на настольный ПК.

Вы считаете, это тот самый телефон, что принесет славу Linux?
Есть и другие претенденты. Что-то может получиться из инициативы Trolltech (телефон Greenphone мы рассматривали в LXF91, а руководство разработчика к нему от Грэма Моррисона из LXF92 записано на сопроводительный DVD нынешнего номера *Linux Format*). Motorola вносит значительный вклад в развитие Linux, а есть и новички, например, ROAD Handy PC (www.road-gmbh.de/englisch/index.html). Но OpenMoko имеет среди них нисколько не худшие шансы.

А вы ничего не забыли?
Что, мамулин день рождения? А, понял, вы имеете в виду Интернет-адреса сайтов, где можно узнать больше. Хорошо. Страниц на SourceForge по этому вопросу нет, поэтому единственное место, где можно получить информацию – это Wiki на сайте OpenMoko. Там собрано немало сведений не только об отдельном проекте, а обо всех аспектах разработки ПО для мобильников. Возьмите стилус и настучите в браузере вашего Linux-смартфона: http://wiki.openmoko.org/wiki/Main_Page. LXF

ПОКА ВЕРСТАЛСЯ НОМЕР

Компания Trolltech объявила о выпуске продукта *Qtopia Phone Edition* для Neo1973 и платформы OpenMoko. Таким образом, сторонние разработчики получили возможность выбора инструментария – *GTK+* (Hildon) или *Qt*. Примечательно, что для этой платформы *Qtopia Phone Edition* был впервые представлен по лицензии GPL (версия 2).



DNS: Связующее звено Интернета

Система доменных имен поистине вездесуща. Она обеспечивает нам комфортную работу в Сети, хотя на нее редко кто обращает внимание в повседневной жизни. **Сергей Супрунов** попытается исправить эту несправедливость.

Четырехбайтная адресация, лежащая в основе IP-протокола, на котором, помимо всего прочего, зиждется сеть Интернет, удобна для операционных систем, но слишком тяжела для слабого и изнеженного человеческого мозга. Поэтому «записная книжка», из которой в любое время можно было бы узнать, что сайт linuxformat.ru нужно искать по адресу 88.212.196.134, а почту для mail.ru отправлять на 194.67.23.20, всегда будет востребована.

Поначалу с ролью такой «записной книжки» превосходно справлялся простой текстовый файл HOSTS.TXT. За его формирование отвечал Сетевой информационный центр Стенфордского института, а администраторы и пользователи периодически скачивали его по FTP. Но по мере роста числа компьютеров, объединенных в единую сеть, такое решение стало абсолютно неэффективным.

В конечном итоге DNS (Domain Name System, система доменных имен) уверенно и, похоже, надолго, заняла место службы, снабжающей нас информацией о том, какой же IP-адрес соответствует интересующему нас доменному имени. Основным преимуществом DNS является ее распределенная природа – информация не концентрируется в одном месте, а разбросана по всему Интернету в соответствии с иерархической структурой пространства доменных имен.

Во главе этой иерархии размещается так называемый корневой домен, обозначаемый одиночной точкой. Из этого корня «растут» домены первого уровня – **ru**, **uk**, **com**, **net**, **org** и т.д. Для реального использования доступны домены, начиная со второго уровня. Внутри домена могут размещаться как отдельные хосты, так и поддомены. Например, www.ibm.com – хост в домене **ibm.com**, а austin.ibm.com – поддомен.

Здесь нужно ввести еще одно понятие – «зона». Зоной называют область пространства имен домена, которая администрируется независимо от остального пространства имен. Жесткой привязки к делению на поддомены здесь нет – поддомен может обслуживаться как автономная единица, т.е. иметь свою зону (в этом случае говорят о делегировании поддомена), а может входить в родительскую зону.

Так как же DNS позволяет узнать IP-адрес того или иного хоста, например, www.yandex.ru? Выглядит это примерно так. Клиент отправляет запрос близлежащему DNS-серверу (обычно это сервер провайдера, но мы увидим далее, что можно запустить и собственный). Этот сервер в общем случае не знает требуемый IP, и даже не представляет, где его искать. Поэтому обращается к одному из корневых DNS-серверов. Корневой сервер физически не в силах держать информацию обо всех доменах третьего (да даже и второго) уровня, так что все, что он может сделать, это послать нас... правильно, к DNS-серверу первого уровня, который обслуживает зону **ru**.

В дела отдельных доменов и этот сервер не вникает, так что адреса конкретных узлов не обслуживает. Зато он знает, какой DNS-сервер отвечает за домен yandex.ru. Туда он нас (точнее, сервер нашего провайдера) и отправит. А вот DNS-сервер домена yandex.ru уже просто обязан вернуть нам IP-адрес входящего в его зону ответственности хоста. Ну или послать... на этот раз просто послать, если искомый хост не существует в природе.

Таким образом, за каждый домен может отвечать свой администратор, настраивая DNS-сервер для его обслуживания, а вышестоящим серверам достаточно знать, что и кому они делегировали. За счет этого и обеспечивается потрясающая масштабируемость, сделавшая службу DNS столь эффективной.

Желание клиента – закон

Начнем наше знакомство с клиентской части. Реализована она в виде стандартной библиотеки, и используем мы ее, можно сказать, ежеминутно, сами того не замечая. Если точнее, то используют ее те приложения, с которыми мы работаем – браузеры, FTP- и почтовые клиенты, различные сетевые серверы, да практически любые приложения, работающие в сети.

Настраивается клиент с помощью файла **/etc/resolv.conf**. В большинстве случаев достаточно указать там DNS-серверы, адреса которых предоставляет вам провайдер:

```
nameserver 1.2.3.4
nameserver 4.3.2.1
```

В современных дистрибутивах эта операция зачастую выполняется через графический интерфейс. Если для настройки сети используется DHCP-клиент, то он, как правило, и осуществляет «редактирование» **/etc/resolv.conf**. Т.е. обычный пользователь сталкивается с данным файлом не часто. Тем не менее, полезно знать, куда заглянуть в случае проблем.

Помимо упомянутой опции **nameserver**, в **resolv.conf** могут задаваться и некоторые другие параметры, но сейчас они нам не слишком интересны. Подробности ищите на странице **man resolv.conf(5)**.

BIND: Узелок на память

Теперь поговорим о сервере. Наиболее распространенным пакетом, обеспечивающим работу службы DNS на Unix-подобных системах, в настоящее время является *BIND* (Berkeley Internet Name Domain). Посмотрим, какая от него может быть польза, даже если вы не располагаете собственным доменом.

Установка сложностей никаких не представляет – даже если *BIND* не включен в состав вашего дистрибутива изначально, его, как правило, несложно найти в официальном репозитории. Скажем, в Ubuntu достаточно одной команды – **sudo aptitude install bind9**, и спустя 300 килобайт трафика один из лучших DNS-серверов в мире будет трудиться вам на благо.

Самое интересное то, что свежеустановленный *BIND* уже готов к работе в качестве кэширующего DNS-сервера. То есть он не отвечает ни за какую зону (за исключением **localhost**), но способен разрешать поступающие к нему запросы, сохраняя ответы в локальном кэше. При повторном запросе ответ уже будет возвращен из кэша, за счет чего достигается некоторая экономия трафика и времени.

Для управления сервером (его исполняемый файл носит название **named**) можно использовать стандартные средства вашего дистрибутива (скажем, сценарий **/etc/init.d/bind9**; в других дистрибутивах он может называться просто **bind** или **named**). Помимо этого, существует специальная утилита **rndc** (замечу, что, поскольку во время работы она взаимодействует с уже запущенным демоном **named**, то запускать его она как раз и не умеет). Чаще всего вы будете сталкиваться с командой **rndc reload [zone]**, позволяющей перезагрузить зону (или все зоны) после внесения изменений. Состояние сервера вернет вам **rndc status**. Введите **rndc** без параметров, и вы узнаете обо всех ее возможностях. Чтобы начать использовать свой DNS-сервер, укажите его IP-адрес в **/etc/resolv.conf**.

Пусть работает трактор в поле...

Как нетрудно догадаться, чтобы получить IP-адрес, соответствующий имени www.yandex.ru, кэширующий сервер будет вынужден итеративно обращаться к различным серверам, начиная с корневого, пока не получит искомое. Для крупного DNS-сервера это вполне нормальный режим работы (заодно пополняющий кэш сведениями о промежуточных серверах), но в нашей небольшой сети мы хотели бы максимально снизить нагрузку и трафик. В случае с *BIND* львиную долю нагрузки можно переложить на плечи DNS-сервера нашего провайдера, указав в конфигурационном файле **named.conf** (в Ubuntu его можно найти в **/etc/bind**) параметр **forwarders** в разделе **options**:

```
options {
    . . . другие опции . . .
    forwarders {
        1.2.3.4; # IP-адрес DNS-сервера провайдера
    };
    // forward only;
};
```

Теперь все запросы, на которые наш сервер не сможет ответить из своего кэша, он будет пересылать DNS-серверу провайдера (можно указать и несколько), ожидая от последнего уже готового ответа. Раскомментировав строку **forward only**, мы вообще запретим нашему серверу самостоятельно искать ответ, даже если ни один из *forward*-серверов наш запрос не удовлетворит.

Раз уж мы здесь, рассмотрим конфигурационный файл **named.conf** чуть подробнее. Синтаксис у него, как видите, «Си-подобный», поэтому не забывайте завершать каждую опцию и блок точкой с запятой. Блок **options** содержит общие для сервера опции, для каждой обслуживаемой зоны должен присутствовать блок **zone** (обратите внимание на зоны, по умолчанию присутствующие в **named.conf**; зона «**.**» – подсказка серверу, где искать корневые серверы, чтобы с чего-то начать).

Своя зона

Что еще можно сделать, имея собственный DNS-сервер? Ну, например, мы можем создать «локальную» зону для наших внутренних узлов. Скажем, чтобы по адресу **webserver.local** открывалась наша внутренняя интернет-страничка, а **ftpserver.local** вел на FTP-сервер. Конечно, можно прописать необходимые соответствия в **/etc/hosts** на каждом узле локальной сети, но зачем так усложнять себе жизнь?

Итак, в **named.conf** добавляем описание нашей «зоны»:

```
zone "local" {
    type master;
    file "/etc/bind/local.db";
};
```

И в файл **/etc/bind/local.db** заносим информацию о наших узлах:

```
$TTL    3d
@       IN      SOA     admin.ns.local. (
                1               ; Порядковый номер
                2d              ; Период обновления
                1h30m           ; Повторение попытки
                1w              ; Устаревание slave-зоны
                1h )            ; Время жизни отрицательных ответов
;
                        IN   NS      ns.local.
ns                      IN   A               192.168.0.254
webserver IN   A                     192.168.0.2
ftpserver  IN   CNAME                webserver.local.
```

В подробности вдаваться не будем – если интересно, ответы на все вопросы вы найдете в замечательной документации (**man 5 named.conf**). Пока достаточно знать, что A-запись ставит в соответствие имя хоста его IP-адресу, а CNAME-запись позволяет указать для хоста дополнительное имя. NS указывает на DNS-сервер, отвечающий за данную зону (т.е. на наш сервер). Полные доменные имена обязательно должны заканчиваться точкой, имена без точки будут дополняться именем зоны. Ну и нужно знать про еще одну запись – PTR, отвечающую за «обратное» разрешение (т.е. поиск доменного имени по IP-адресу), для которого нужно создать еще и *in-addr.arpa*-зону:

```
zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/0.168.192.in-addr.arpa.db";
};
```

Соответствующий файл **0.168.192.in-addr.arpa.db**:

```
$TTL    3d
@       IN      SOA     admin.ns.local. (
                1               ; Порядковый номер
                2d              ; Период обновления
                1h30m           ; Повторение попытки
                1w              ; Устаревание slave-зоны
```

»

Скорая помощь

Если, несмотря на все шаманство с **resolv.conf**, разрешение имен идет как-то не так, загляните в **/etc/nsswitch.conf**, в строку hosts. Там указан порядок поиска доменных имен: **files** означает файл **/etc/hosts**, **dns** – поиск с помощью DNS-клиента. Возможно, тут и кроется проблема?



»
Маленькие шалости с DNS

Думаю, вы уже поняли, что собственный DNS-сервер позволяет и немножко пошалить. Например, что помешает нам создать *master*-зону **microsoft.com**? Вот наши пользователи удивятся, когда вместо www.microsoft.com попадут на linux.org! Главное – не переусердствовать, ведь потерять доверие других людей – это уже совсем не шутки...

```
            1h )        ; Время жизни отрицательных ответов
;
                        IN NS              ns.local.
2                       IN PTR    webserver
254                     IN PTR    ns
```

Фигурирующую в самом начале SOA-запись можно, не вдаваясь в особые подробности, скопировать из какого-нибудь файла-примера – в ней задаются преимущественно различные таймауты, и значения по умолчанию обычно неплохо подходят. Точка с запятой начинает комментарий.

На всякий случай замечу, что файлы зон вы вольны называть как душе угодно. Просто для удобства принято, чтобы имя файла соответствовало обслуживаемой зоне: сразу видно, за что именно тот или иной файл отвечает, и не обязательно сверяться с **named.conf**.

Теперь машины нашей локальной сети будут получать нужные адреса при запросах к «зоне» **local** (при условии, что в настройках в качестве DNS-сервера указан наш), ну а запросы к другим зонам будут обслуживаться как в случае кэширующего сервера.

Служить бы рад...

Рассмотрим еще один случай. Предположим, что пользователи вашей небольшой локальной сети активно работают с ресурсами, имена которых обслуживаются зоной вашего провайдера. Безусловно, использование кэширующего DNS-сервера поможет разгрузить и DNS-сервер провайдера, и ваш интернет-канал. Однако можно поступить еще лучше – настроить свой *BIND* в качестве slave-сервера для зоны провайдера. Подчиненный (slave) сервер DNS предназначен для резервирования основного сервера, т.е. он тоже занимается обслуживанием зоны, но с той разницей, что файл зоны он берет не с диска, а скачивает с основного (master) сервера. Если данные этой зоны меняются не часто, то, единожды загрузив файл зоны (выполнив трансфер зоны), вы сможете самостоятельно обслуживать запросы к ней без обращений к DNS-серверу провайдера. Описывается slave-зона столь же легко:

```
zone "your.provider.ru" {
            type slave;
            file "/var/bind/slaves/your.provider.ru.db";
            masters { 1.2.3.4; }; # IP-адрес провайдерского DNS-сервера
};
```

После перезагрузки сервера (например, командой **rndc reload**) у вас должен появиться указанный файл. Вручную создавать его не нужно; только позаботьтесь, чтобы каталог **/var/bind/slaves** принадлежал пользователю *BIND* или, как минимум, был доступен ему на запись. Теперь ваш сервер сможет авторитативно (то бишь компетентно, официально) отвечать на запросы, касающиеся зоны **your.provider.ru**, без дополнительных обращений к серверу провайдера. (Нужно заметить, что зачастую администраторы DNS-серверов по соображениям безопасности запрещают передачу зоны на произвольные узлы. Если это ваш случай, то попросите провайдера разрешить трансфер для вашего адреса – в конце концов, он тоже заинтересован, чтобы его сервер не беспокоили по пустякам.)

Кстати, то, что мы получили, на самом деле не настоящий slave-сервер, поскольку ни одна из NS-записей зоны **your.provider.ru** на него не ссылается. Так что «чужие» о нашем сервере ничего не узнают и не смогут им пользоваться – а нам оно и не надо... Если же вы настраиваете полноценный slave-сервер для делегированного вам поддомена, то укажите его в файле зоны в качестве NS-сервера наряду с основным:

```
@           IN NS        ns.mydomain.ru.
            IN NS slave.mydomain.ru.
```

Теперь клиенты будут обращаться случайным образом как к основному, так и к подчиненному серверу, распределяя нагрузку. А в случае выхода из строя основного, *slave*-сервер, как ему и положено, возьмет обслуживание зоны на себя.



Другая точка зрения

Допустим, у вас есть делегированный вам домен **mydomain.ru**, который обслуживается вашим DNS-сервером. И здесь может возникнуть две интересные задачи. Во-первых, иногда нужно скрыть некоторую информацию от любопытных глаз, предоставляя ее лишь избранным. Например, ваша компания оказывает услуги доступа в Интернет, и вы хотите, чтобы ваш FTP-сервер посещали лишь ваши абоненты. (Да, это делается настройкой самого FTP-сервера, но было бы неплохо, чтобы посторонние о нем даже и не знали...) Можно ли предоставить адрес вашего сервера только пользователям конкретного домена?

Во-вторых, в локальной сети, как правило, используются «серые» IP-адреса, и обращение к «внешним» по отношению к ней ресурсам связано с трансляцией адресов. Но если ресурс работает на машине с несколькими сетевыми интерфейсами и доступен в том числе и по «серому» адресу, то такая трансляция уже несколько избыточна. Можно ли возвращать пользователям локальной сети «серый» адрес в ответ на запрос, не вынуждая их использовать альтернативные имена типа **inner.mydomain.ru** или тот же «домен» **local**?

В случае использования *BIND9* ответы на оба вопроса будут положительны. В нем появилась одна замечательная штука – оператор **view**. Он позволяет разграничить «виды» одной и той же зоны для разных групп клиентов. Выглядит это примерно так:

```
view "clients" {
   match-clients { 1.2.3/24; };
    zone "mydomain.ru" {
        type master;
        file "/etc/bind/mydomain.clients.db";
    };
    . . . прочие зоны . . .
};

view "others" {
   match-clients { any; };
    zone "mydomain.ru" {
        type master;
        file "/etc/bind/mydomain.db";
    };
    . . . прочие зоны . . .
};
```

Такими настройками мы выделили два «вида» – для наших клиентов (сеть 1.2.3.0/24) и всех остальных. Осталось лишь подготовить различные файлы одних и тех же зон для разных «видов», и задача будет решена. Обратите внимание, что если вы приняли решение использовать «виды», то не должно быть ни одного оператора **zone**, не входящего в какой-нибудь оператор **view**.

dnsmasq: не BIND'ом единым...

BIND является одним из самых мощных и настраиваемых DNS-серверов. Однако порой его возможности выглядят излишними, а кого-то могут и испугать...

Конечно, *BIND* нельзя назвать ресурсоемким приложением, особенно применительно к современному оборудованию. Да и настройки не столь обременительны, в чем я, надеюсь, вас убедил. Однако, если единственной задачей, возлагаемой на локальный DNS-сервер, является кэширование запросов пользователей, то для ее решения можно воспользоваться и более специализированным сервисом.

Одним из таковых является кэширующий DNS-сервер *dnsmasq* (по совместительству он может выполнять и роли DHCP-сервера, раздавая машинам локальной сети IP-адреса и прочие сетевые настройки, необходимые для работы, и TFTP-сервера для обеспечения работы бездисковых станций, загружающихся по сети). Его установка выполняется столь же легко, как и *BIND* – многие дистрибутивы уже включают его в базовой поставке, если нет – почти наверняка вы найдете его в официальном репозитории. Для настройки используется файл **/etc/dnsmasq.conf** (в других дистрибутивах место расположения может быть иным).

То, что конфигурационный файл содержит свыше 400 строк, не должно вас пугать – большую их часть составляют довольно подробные комментарии, да и значимые строки отвечают преимущественно за работу сервера DHCP. Основные же DNS-параметры сосредоточены в начале файла.

На что следует обратить внимание в настройках? Во-первых, *dnsmasq* руководствуется файлом **resolv.conf**, определяя, каким DNS-серверам следует перенаправлять запросы, которые не могут быть удовлетворены из кэша. При необходимости вы можете определить другой файл вместо **/etc/resolv.conf** в параметре **resolv-file** либо запретить его использование, раскомментировав параметр **no-resolv**. Параметрами **server** можно указывать «вышестоящие» DNS-серверы прямо в конфигурационном файле. Причем вы можете даже задавать различные серверы для отдельных доменов:

```
no-resolv
server=/provider.ru/1.2.3.4
server=4.3.2.1
```

Эти строки предписывают *dnsmasq* игнорировать сведения из **/etc/resolv.conf** и использовать сервер 1.2.3.4 для разрешения имен домена **provider.ru**, и сервер 4.3.2.1 для всех прочих запросов.

Помимо выбора forward-серверов, вы можете указать пользователя и группу, от имени которых будет выполняться *dnsmasq*, конкретизировать интерфейсы, на которых *dnsmasq* будет ожидать запросы от пользователей и т.д.

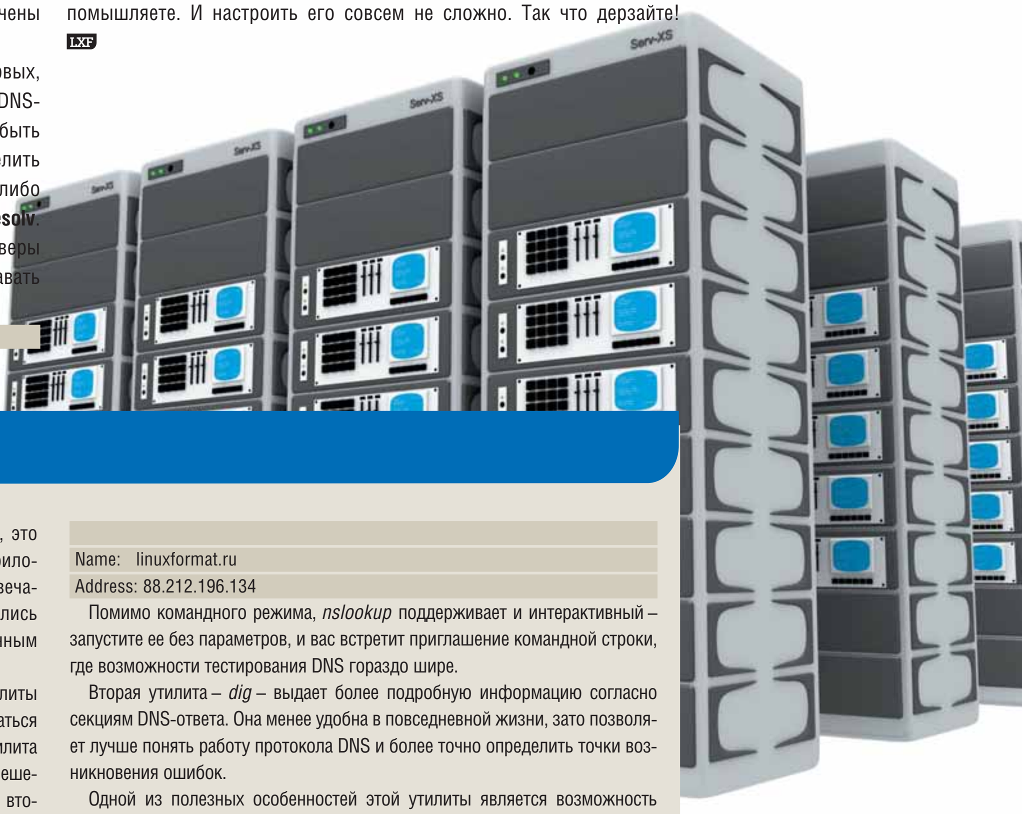
Еще одна полезная особенность – по умолчанию *dnsmasq* при старте читает файл **/etc/hosts** и заносит найденную там информацию в свой кэш. Благодаря этому можно обеспечить разрешение локальных имен без необходимости синхронизировать **/etc/hosts** на всех машинах локальной сети. Если вам эта функция не нужна, отключите ее опцией **no-hosts**.

Помимо редактирования файла **dnsmasq.conf**, практически все параметры работы можно задать непосредственно в командной строке. Подробности – на man-страницах.

Как видите, собственный DNS-сервер может оказаться весьма полезной (ну, как минимум, интересной) штукой, даже если в вашей локальной сети всего пара машин, а о собственном домене вы и не помышляете. И настроить его совсем не сложно. Так что дерзайте! **LXF**

Скорая помощь

Если ваш DHCP-клиент ISC при каждой перезагрузке затирает **resolv.conf**, раскомментируйте строку «**prepend domain-name-servers 127.0.0.1;**» в **/etc/dhcp3/dhclient.conf**. Вместо 127.0.0.1 укажите IP-адрес своего DNS-сервера.



Другая точка зрения

DNS работает как бы сам по себе, но если начинаются проблемы, это сказывается самым неожиданным образом на самых различных приложениях: от браузера до сервера электронной почты. Пока за DNS отвечает ваш провайдер, беспокоиться особо не о чем, но раз уж мы взялись сами за настройки этой службы, то нужно быть готовым к различным сюрпризам.

Традиционно для тестирования работы DNS используются утилиты *nslookup* и *dig*. Первая издавна шла в составе пакета *BIND*, и пользоваться ею очень просто – укажите в качестве аргумента доменное имя, и утилита вернет IP-адрес и сведения о сервере, который использовался для разрешения имени. При необходимости вы можете указать любой DNS-сервер вторым аргументом, которому будет отправлен запрос:

```
amsand:~$ nslookup linuxformat.ru ns.mezon.ru
Server:       ns.mezon.ru
Address:      83.68.34.21#53

Name:   linuxformat.ru
Address: 88.212.196.134
```

Помимо командного режима, *nslookup* поддерживает и интерактивный – запустите ее без параметров, и вас встретит приглашение командной строки, где возможности тестирования DNS гораздо шире.

Вторая утилита – *dig* – выдает более подробную информацию согласно секциям DNS-ответа. Она менее удобна в повседневной жизни, зато позволяет лучше понять работу протокола DNS и более точно определить точки возникновения ошибок.

Одной из полезных особенностей этой утилиты является возможность задать в переменной окружения *LOCALRES* путь к альтернативному файлу вместо **resolv.conf**. Эта переменная никак не влияет на работу других программ, так что с помощью *dig* можно проверить работоспособность нового файла либо использовать альтернативный в особых случаях.



Мнение: Джереми Эллисон

Неподвластное времени

Джереми Эллисон

Джереми – ведущий разработчик команды *Samba*, сейчас работает в Google, в Сан-Хосе, Калифорния.

Недавно в Тулсе, Оклахома, была извлечена из земли памятная капсула 1957 года. Ее основным содержимым был автомобиль — Плимут Бельведер 1957 года. При закладке капсулы его предусмотрительно снабдили канистрой бензина, сочтя, что летающие автомобили 2007 года уже не будут нуждаться в столь ископаемом топливе. В сопроводительных бумагах к автомобилю, последний именовался «бесценным для 2007 года антиквариатом»!

Безусловно, это антиквариат, но не очень-то бесценный. В капсулу просочились грунтовые воды, и бедняге подпортило ржавчиной корпус, крылья, да и все в целом. Нам, умудренным людям 2007 года, легко смеяться над наивностью создателей капсулы, однако разве мы сами лучше, когда оставляем историкам будущего свои письменные свидетельства?

Все наши письменные свидетельства быстро переходят на электронные носители. Хорошим примером тому служит закат популярности фотопленки – почти никто в наши дни не покупает пленочных фотоаппаратов; фотографии теперь хранятся на картах памяти или загружаются на крайне хрупкие жесткие диски компьютеров, обычно без всяких резервных копий. Даже будучи профессиональным компьютерщиком, я проявляю не меньшую лень по части резервного копирования фото, чем все прочие. Одна несчастная случайность – и все мои драгоценные воспоминания будут безвозвратно утрачены.

> «Кто знает, какие еще открытия появятся по мере расширения доступа к знаниям человечества.»

Недавно в NASA обнаружили, что оригинальные пленки медленной развертки с первой высадки человека на Луну с Аполлона-1 пропали. Те кадры, что видели все, на самом деле лишь копия тех пленок, снятая с видеозаписи с высоким разрешением телекамерой 1969 года. Согласно сообщениям NASA, у них имеется 2 612 коробок с пленками, однако не хватает еще 13 000 дополнительных пленок. Это мне напоминает склад, который показывают в конце «Поисков утраченного ковчега» [Raiders of the Lost Ark – один из фильмов об Индиана Джонсе, – *прим. пер.*].



Даже если эти пленки найдутся – интересно, много ли сохранилось киноаппаратов образца 1969 года в рабочем состоянии, чтобы можно было воспроизвести эти данные? Насколько обветшает к тому моменту пленка? И это всего лишь за 40 лет – или около того. А что будет через 100 лет, или через 500, или 1000? Пленки со съемками высадки на Луну – историческая ценность, и если мы о них не можем позаботиться, то что уж говорить о менее важных записях? Письменные записи, по крайней мере, можно расшифровать и через много тысяч лет.



Я вовсе не луддит: преимущества перевода наших данных и записей в цифровой формат серьезно перевешивают все возможные недостатки. К примеру, оцифровка всех книг и данных музеев сделает их доступными для тех ученых, у кого есть доступ в Интернет, но нет шансов прикоснуться с этими данными в реальности. Да и вообще, возможность в информационных массивах весьма привлекательна. Сейчас, при свободном доступе к спутниковым снимкам через Интернет, стали неожиданно делаться важные открытия – например, метеоритных кратеров. Кто знает, что еще откроется благодаря переводу человеческих знаний в общий доступ? Однако было бы неплохо задуматься о сохранении возможности доступа к важным историческим записям нашего времени.

Полагаю, проприетарные форматы записи станут проблемой для историков. Возможно, не в ближайшем будущем – но уж совершенно точно, через некоторое время (я говорю о сотнях, если не о тысячах лет). Представьте себе, как лет через 500 некий историк обнаруживает «засекреченный бункер» вице-президента Чейни, где спрятан его личный ноутбук. «Ура, – думает этот историк, – наконец-то мы узнаем, кто же давал советы этой администрации по поводу нефтяной политики!» – и вновь погружается в океан, поглотивший бывший Вашингтон. И тут оказывается, что данные записаны в формате «Word-Кривой для Windows 2002», который специально создавался так, чтобы конкуренты не могли его прочесть.

Кроме шуток, проприетарные форматы усложняют задачу сохранения нашей культуры почище устаревших аппаратных интерфейсов и подверженности носителей данных разрушению, хотя мы привыкли считать их вечными и незыблемыми. Использовать форматы данных, не являющиеся опубликованными стандартами, означает просто напрашиваться на неприятности при долгосрочном хранении данных. Я отдаю предпочтение формату ODF, формат Microsoft Office Open XML (OO-XML) тоже документирован, так что он не вызовет проблем при долгосрочном хранении данных. Однако большинство документов мира – как на правительственном уровне, так и на корпоративном – по-прежнему сохраняются в проприетарных форматах, а ведь порой документы, считавшиеся недостойными сохранения, оказываются самыми желанными для историков.

Меня не слишком волнуют «сценарии судного дня», рисующие постиндустриальную, лишенную электричества цивилизацию, утратившую все научные знания по причине неспособности прочитать документы, созданные на компьютере: когда общество в таком положении, у него есть более серьезные поводы для беспокойства, чем невозможность получить доступ к древним финансовым документам предприятий. Нет, меня беспокоит другое: вдруг кто-то захочет увидеть старую пленку с записью падения Берлинской Стены, и обнаружит, что есть только копии с копий с других копий, а оригинальная «правда» этого события утрачена сотни лет назад. Или, что еще хуже, не сможет определить, которая из версий этого видео показывает настоящее событие.

Мой любимый мультипликатор Тед Ролл [Ted Rall] (**www.rall.com**) использует 2024 как римейк оруэлловской антиутопии «1984». Герой от нечего делать редактирует список лауреатов Нобелевской премии Мира и подсовывает в него имя местного рок-музыканта. Средства массовой информации получают и распространяют этот список, а герой уже позабыл о своей шалости и принимает подаваемую информацию на веру, как и все остальные. Когда живые очевидцы события исчезнут, и сохранятся только электронные свидетельства, с присущей им подверженностью изменениям, узнают ли люди правду о нашем прошлом? Возможно, DVD или запись на ином физическом носителе, точно привязанном к хронологии исторических событий, может стать ценнее, чем любые данные в электронном виде, и даже послужит арбитром для имеющихся цифровых версий. Возможно, мы оставляем грядущим поколениям проклятие в виде такого неимоверного количества версий одних и тех же данных, что невозможно будет отличить, где же среди них оригинальная правдивая версия, если она вообще есть.

И, напоследок, не могу удержаться от комментария в адрес NASA – они ответственны за сохранение единственного долгосрочного носителя данных, когда-либо созданного человечеством. Космические корабли Пионер и Вояджер несут на борту плакетки с указанием расположения нашей солнечной системы; на Вояджере также находится диск с аудио- и видеозаписями с Земли. Есть надежда, что миллионы лет спустя далеко за пределами Солнечной системы эти послания достигнут внеземного разума, который сможет прочесть их. Вот памятная капсула, достойная восхищения! LXF

Изображения планет взяты из *Stellarium*, открытого программного планетария для Linux, Windows и Mac OS X на OpenGL. www.stellarium.org

LINUX FORMAT

Учебники



Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

GNU/Linux в «большой» политике?

— Проблемы? спросил Ваймс, догоняя быстро семенившего Ди.

— У нас нет никаких проблем.

«Ага, он солгал, — подумал Ваймс. — Начинается дипломатия».

Терри Пратчетт. Пятый слон

Я зарёкся упоминать в своих колонках сильных мира сего – их цели, как правило, сильно отличаются от целей остального человечества. Но упомянуть о как минимум поверхностной метаморфозе, произошедшей с Мининформсвязью, управляемой Л. Д. Рейманом, всё-таки, наверное, стоит.

Менее полугода назад это министерство отличилось совершенно бездарным планом закупок лицензионного программного обеспечения для школ и заслуженно получило «оплеуху» от ведомства Грефа. К сожалению, автор «оплеухи» Андрей Шаронов уже уволился из Минэкономразвития, но как следствие, Мининформсвязь породила проект, задачей которого является «разработка дистрибутива общесистемного и офисного программного обеспечения на основе свободного программного обеспечения для образовательных учреждений Российской Федерации» – то есть правильные слова уже выучили.

Совершенно не очевидно, что из этого получится что-то полезное, поэтому расслабляться не стоит, но и хуже, чем сейчас, явно не будет. Тем более, если вернуться к теме сильных мира сего, то одному из них фирма IBM сумела навязать ноутбук с GNU/Linux, и он уже был неоднократно замечен в использовании этого продукта. Подробности можно прочитать по адресу http://abbra.livejournal.com/94656.html. Это ни о чём особо не говорит, но информация о GNU/Linux неудержимо расползается. Как говорится: мелочь – а приятно.

P.S. Исходные тексты и графика игры AstroMenace (LFX95) теперь доступны под лицензией GPL3. С моей точки зрения, это уже отнюдь не мелочь.

P.P.S. SCO объявила себя банкротом. Вот уж действительно мелочь – но всё равно приятно.

E.M.Baldin@inp.nsk.su

В этом выпуске...

Совет месяца: rename и prename

Едва ли отыщется пользователь командной строки, не знакомый с командой **mv**. Зачастую это самый быстрый способ привести файлы в порядок, хотя нажатие F2 в современном файловом менеджере позволит вам сделать то же самое, но графически. Буквы «mv» являются сокращением от глагола «move» (перемещать), куда более вразумительно описывающего действие команды. Действительно, имя файла может включать и путь, поэтому, набрав **mv /usr/sbin/sux /sbin/sux**, вы измените не название, а местоположение. Переименовывать файлы в прямом смысле этого слова тоже можно, например, при размещении их в резервном каталоге нелишне будет добавить и расширение «**_bak**». Mv хорошо справляется с одиночными файлами или директориями, но пасует, когда речь идет о групповом переименовании.

К счастью, небезызвестный Ларри Уолл [Larry Wall] создал еще одну команду, которую можно отыскать практически в любом дистрибутиве Linux. Обычно она называется **rename**, но вам следует поискать prename или что-то похожее, если имя «rename» уже занято в вашей системе под что-то другое. Естественно, уж коль скоро к этому приложил руку Ларри, **rename** – это сценарий Perl, так что вы можете открыть его в своем любимом текстовом редакторе. Основное предназначение данного скрипта – легко перемалывать группы файлов. Однако его мощь зиждется на анализаторе выражений Perl, которые не всегда просты для понимания, особенно если у вас нет опыта. Например, чтобы добавить расширение «**_bak**» к каждому файлу, нужно набрать следующее:

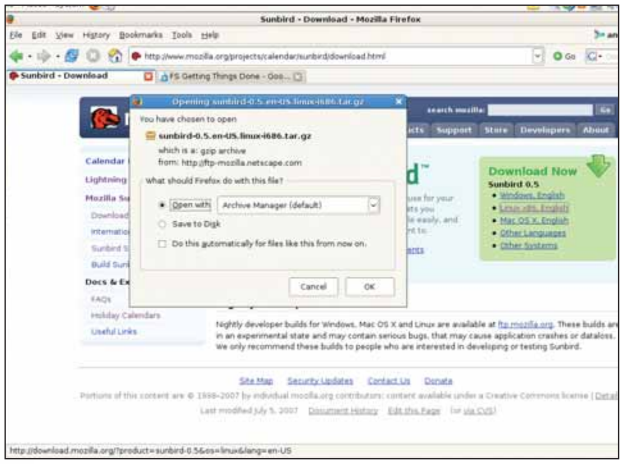
```
prename 's/\_bak$//' file*
```

Это только начало – после того, как вы овладеете механизмом регулярных выражений Perl, вы сможете применять **prename** для решения практически любых задач. Он особенно удобен при переименовании файлов изображений, загруженных с раздела Windows, из верхнего регистра в нижний.

Органайзер:

Лекарством для недугов наших дней (кроме снижения затрат ископаемого топлива) была бы лучшая организация. **Энди Ченнел** – хозяин своего времени...





› *Sunbird* **имеется во многих основных рабочих столах. Linux-версия – это архив .tar.gz, с которым ваш ПК должен справиться автоматически. Если нет, дважды щелкните на нем после загрузки.**



Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется еще со времен Dragon 32

Темп жизни, похоже, не собирается замедляться, и потерянное искусство тянуть время становится, хм, все более потерянным. Но так не обязательно должно быть. При содействии некоторых стандартных программ Linux и правильного планирования все еще возможно (по крайней мере, иногда) спихнуть на завтра то, что нужно сделать сегодня. На самом деле, без этого не обойтись!

Ключом к продуктивным проволочкам является умение точно представить, когда дело должно быть сделано, сколько времени оно займет и что важнее. А ключевые типы программ, помогающие этого добиться – Календарь и Список задач [To Do]. К счастью, в Linux есть прекрасные программы для обоих рабочих столов, Gnome и KDE, в лице *Evolution* и *Kontact* соответственно, но существуют также и кросс-платформенные пакеты, вроде *Sunbird* – и его дочернего проекта *Lightning*, интегрированного в *Thunderbird* – а также целый диапазон онлайн-решений, идущих ноздря в ноздрю с настольными соперниками. Ради простоты мы выбрали для данного урока *Sunbird/Lightning*, но приведенные здесь методы подходят к практически любому мыслимому приложению, управляющему расписанием. *Sunbird* делает все необходимое для контроля за нашим временем; мы создадим календари для задач, добавляя и редактируя события, установим напоминания, подписавшись на сетевые списки событий, синхронизируемся с нашим сетевым календарем и расставим приоритеты работ, настроив список задач. Через страницу мы также покажем, как идти в ногу со временем вдали от вашего Linux-компьютера при помощи новаторского (и очень дешевого) органайзера *Graphite*.

Кое-что из этого основано на идее Getting Things Done (GTD) Дэвида Аллена [David Allen] (**www.davidco.com**): как наиболее эффективно управлять информацией и временем [в русском переводе: Д. Аллен «Как привести дела в порядок. Искусство продуктивности без стресса», Вильямс, 2007 г.]. И хотя вся GDT-«индустрия» полнится рекламной шумихой и почти непознаваемым корпоративным сленгом, по сути она выливается в несколько здравых идей, которые можно реализовать, не прибегая к дорогому ПО, книгам или интенсивным семинарам.

Получаем программы

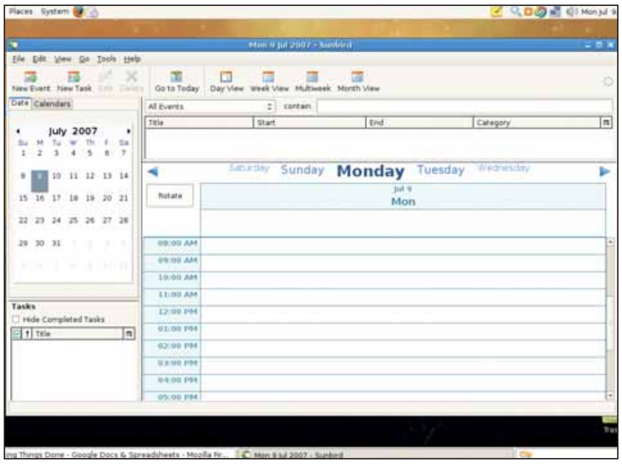
Для начала нам необходимы две программы, первая из которых – сама *Sunbird*. На момент написания была доступна версия 0.5, на **www.mozilla.org/projects/calendar/sunbird/**, она имеется на DVD этого месяца. Загрузив файл, дважды щелкните на архиве и распакуйте его куда-нибудь, где у вас хранятся приложения или двоичные файлы (в нашем случае, это **/home/downloads/имя_приложения**). Затем сам архив можно удалить. Теперь просмотрите каталог *Sunbird* и найдите элемент с именем *Sunbird* (вообще-то это скрипт) и дважды щелкните на нем. Перед вами появится окно с выбором: показать файл, запустить его в терминале или просто запустить. Выберите Запустить [Run] и подождите пока приложение запускается. При первом запуске вам предоставится возможность импорта данных календаря из других источников, включая *Evolution*, а затем перед вами предстанет само приложение, по умолчанию отображающее вид День. Абсолютно понятный набор других видов, доступный в верхнем меню окна, включает Месяц [Month] и Неделю [Week].

Прежде чем двинуться дальше, сделаем на рабочем столе ярлык для этого приложения и заставим его запускаться при каждой загрузке системы – это удобный способ известить себя о насущных задачах на день: календарь должен быть первым, что вы видите.

» **Месяц назад** Мы редактировали видео под Linux – прямо в web-браузере.

Успеть все!

Linux Format На DVD
» Sunbird 0.5
» Lightning 0.5
» Provider 0.2.1



› By default, *Sunbird* launches in Day View, which is ideal for seeing what the coming workday holds.

Итак, в Gnome щелкните где-нибудь на рабочем столе и выберите Создать кнопку запуска... Добавьте соответствующее Имя, затем щелкните по кнопке Просмотреть... рядом с областью Команда. Перейдите к элементу *Sunbird*, на котором вы щелкали ранее, и выберите его. Теперь щелкните по кнопке с текстом Нет значка, выберите Просмотреть и найдите каталог /Icon внутри каталога *Sunbird*. Выберите большую из двух иконок и нажмите кнопку OK. Можете также добавить подсказку для элемента, поместив текст в строку Примечание. Если иконка получилась слегка великоватой для вашего рабочего стола, то щелкните на ней правой кнопкой мыши, выберите Растянуть значок, а затем измените его размер, используя один из четырех квадратиков по углам иконки. Теперь нажмите Система > Параметры > Сеансы и выберите вкладку Запускаемые при старте программы. Нажмите Создать, задайте имя и, уже в последний раз, Просмотреть для перемещения в соответствующее место, чтобы выбрать приложение *Sunbird*.

Процедура для KDE весьма похожа, за исключением того, что после правого щелчка при создании ярлыка на рабочем столе следует выбрать Ссылка на приложение... (и далее то же). Чтобы приложение запускалось автоматически, откройте ваш каталог в **/home**, выведите скрытые файлы (Вид > Показывать скрытые файлы), перейдите в **.kde/Autostart** и создайте в нем ссылку на приложение *Sunbird*.

Теперь, когда бы вы ни включили свой ПК, *Sunbird* в Gnome или KDE будет запускаться автоматически, и вы точно будете знать, чему посвятите свой день – конечно, если позаботитесь о вводе этих данных.

Вторая необходимая нам программа также есть на DVD. Она называется *Provider* и позволяет *Sunbird* (или *Lightning*) осуществлять двустороннюю синхронизацию с календарем Google. Это значит, что если вы обновите календарь *Sunbird*, то эти изменения попадут и в ваш сетевой календарь, и наоборот; а значит, открываете ли вы свои данные в домашнем уюте или в интернет-кафе в Москве, вы всегда будете иметь верную информацию под рукой. Последняя версия доступна на **https://addons.mozilla.org/en-US/sunbird/addon/4631**. Однако если вы просто щелкнете по кнопке Установить [Install Now], ничего не произойдет, потому что расширение разработано для *Sunbird*, а ссылка работает только в *Firefox*. Вместо этого щелкните на ссылке правой кнопкой мыши и сохраните ее на вашем рабочем столе. Теперь откройте *Sunbird*, выполните Инструменты > Дополнения и внизу слева нажмите кнопку Установить... Откроется стандартный диалог открытия файла, и вы сможете выбрать в нем загруженный XPI-файл. По завершении установки понадобится перезапустить *Sunbird*.

Добавляем содержимое

Итак, мы справились с установкой, но пока данных нет, проку от этого мало. Первое, что мы добавим – праздники своей страны: это легко, благодаря тому, что web-сайт проекта в разделе Docs & Extras имеет набор календарей праздников. Отыщите ссылку на вашу страну и сохраните файл на свой жесткий диск. Вернитесь в *Sunbird*, выполните Файл > Открыть календарь... [File > Open Calendar File...] и переместитесь к только что загруженному файлу. Теперь, выбрав вид Месяц, вы должны увидеть, что календарь заселен различными событиями. Можете отключить эти события, перейдя на вкладку Календари [Calendars] (слева от главного окна) и сняв галочку напротив RussiaHolidays. Поскольку *Sunbird* использует распространенный формат файлов-календарей ICS, можно загружать из сети всевозможные варианты – например, игры сборной Манчестера на сезон–2007/2008, и это будет отдельный календарь внутри приложения.

Собственные события можно добавить в календарь, дважды щелкнув на дате или во временной секции любого из видов и добавив соответствующую информацию. Каждое событие должно иметь дату, время и продолжительность (если только это не событие на весь день), и выбранный Календарь (то есть частью какого календаря оно является). Среди других опций, пригодится флажок Повторять, который также позволяет повторить событие согласно определенному шаблону, если, например, надо учесть событие, происходящее в 10:00, но только по будням и только до 22 декабря. После определения события оно тут же появится в соответствующем окошке, подкрашенное цветом, выбранным для данного календаря. Вы можете изменить детали события, дважды щелкнув на нем; или, если необходимо изменить только дату (или время в виде День), можете перенести выбранную секцию в другое место.

Для грамотного управления временем необходимо изучить две основные концепции: разделение и приоритет. Примером первой является собрание всех российских праздников в один календарь *Sunbird*. Эта информация может быть включена или выключена, то есть при желании вы можете полностью ее игнорировать, пока она снова не понадобится. Вторая связана с использованием списков задач с приоритетами – *Sunbird* их тоже поддерживает, но мы воспользуемся сетевым сервисом: он более гибок и прекрасно интегрируется с *Sunbird*.

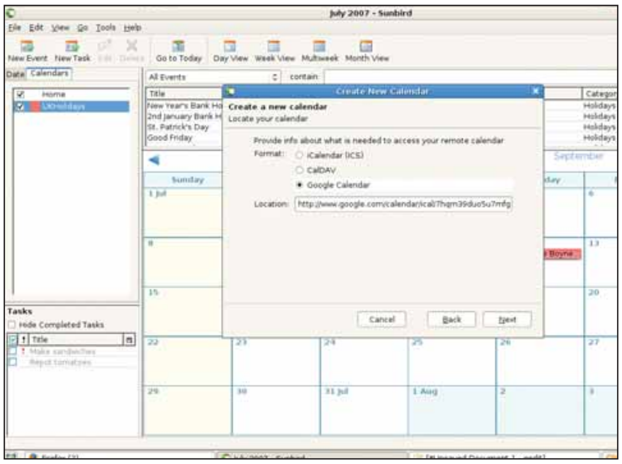
Предположим, в вашей жизни есть три главных приоритета: семья, работа и дела общественные; последнее – в форме заведования школой. Надо также отслеживать важные даты релиз-цикла Fedora. На все это требуется пять календарей, включая уже имеющийся календарь российских праздников. Конечно, можно сгрести все эти данные в один »

Скорая помощь

Работая на разных ПК с разными ОС, используйте кросс-платформенные пакеты, чтобы не гадать, как же это все работает.



› **Если ваш календарь запускается автоматически, значит, вы уже не избежите необходимости соблюдать сроки.**



› **Модуль расширения *Provider* для *Sunbird* позволяет достаточно просто подцепить календарь Google.**

» календарь, но смысл отдельных календарей в том, что вы можете предвидеть свою жизнь, если хотите – то в виде баланса работа/жизнь, на месяц вперед и, если необходимо, полностью «отключить» какую-то из линий, если требуется сосредоточиться на другом.

Первым делом надо переименовать домашний календарь по умолчанию – Мой календарь: он идеально подойдет как календарь общественных мероприятий, поскольку к нему не нужен удаленный доступ. Щелкните правой кнопкой мыши на Моем календаре, выберите Свойства (или дважды щелкните на нем) и введите новое имя. Здесь вы можете также задать новый цвет календаря.

Для создания нового календаря, щелкните правой кнопкой мыши в панели календарей (слева в главном окне) и выберите Новый календарь... или просто выполните двойной щелчок где-нибудь в этой панели для запуска мастера нового календаря. Вначале вам предложат два варианта: календарь на вашем компьютере или в сети? Первый просто хранится на вашем компьютере (так же, как переименованный Общественный календарь), а второй для получения информации соединяется с календарем в Интернете; однако для доступа или изменения данных сетевого календаря не обязательно подключаться к сети, поскольку *Sunbird* создаст локальное зеркало на то время, пока вы отсоединены. Вы можете обновить все позднее, щелкнув правой кнопкой мыши на календаре и выбрав Обновить удаленные календари или используя комбинацию клавиш Ctrl+R.

Скорая помощь

Разбиение больших событий на небольшие части – важный процесс для управления временем.

Нас интересует возможность работать с нескольких разных компьютеров, но так, чтобы иметь доступ к данным и в отсутствие сети, поэтому создадим сетевой календарь. На самом деле, сперва мы подключим существующий календарь Google, а потом – календарь в формате *iCal*. Это означает, что нам не придется вновь вводить кучу данных или, что более важно, потерять их, если жесткий диск выйдет из строя или вор-линуксофил стырит ноутбук LXF: оба аспекта очень важны при решении вверить свои данные какому-либо устройству.

Чтобы внести информацию в *Sunbird*, вначале войдем в службу Google Calendar и выберем календарь, к которому собираемся подключиться. Как и в *Sunbird* (и практически везде), список доступных календарей отображается слева. Щелкните на стрелке рядом с именем календаря и выберите Calendar Settings [Настройки календаря]. Интересующий нас раздел внизу окна справа и озаглавлен Private Address [Личный адрес]. Щелкните правой кнопкой мыши на кнопке XML и выберите Copy link location [Скопировать ссылку]. Теперь вернитесь в *Sunbird* и запустите мастер нового календаря. Выберите календарь В сети и опцию Google Calendar и, по запросу, вставьте скопированный URL из буфера обмена при помощи Ctrl+V. Нажмите Далее. Теперь вам предоставляется возможность задать календарю имя (мы выберем Работа), а также цвет отображения. Цвета очень важны с точки зрения организации: они сразу же наглядно покажут, чем будет занят ваш день, неделя или месяц и насколько вы будете заняты; так что выберите цвет, некоторым образом соответствующий целям календаря, и убедитесь, что отличие в оттенках каждого из них бросается в глаза. В следующем окне необходимо ввести детали для входа на сервер и выбрать, следует ли сохранить эти детали, чтобы календарь мог обновляться автоматически. Когда вы дойдете до конца, данные будут загружены, и, как по волшебству, появятся в *Sunbird*. Теперь сделайте то же для остальных ваших личных календарей. Поскольку мы установили для *Sunbird* модуль расширения *Provider*, то любые изменения, сделанные нами в календарях, связанных с сервисом Google, будут отражены в версии, размещенной в сети.

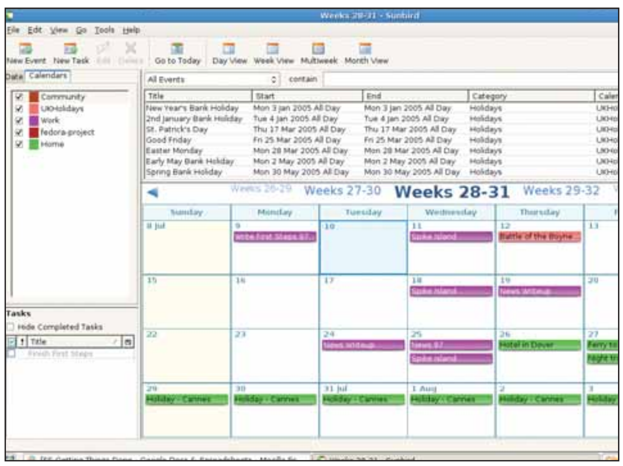
Далее, добавим динамический календарь событий, сосредоточенный на проекте Fedora: вновь запустим мастера нового календаря и выберем опцию В сети и формат iCalendar (ICS). Календарь находится на **webcal://fedora.redhat.com/participate/schedule/fedora-project.ics**, так что вставьте этот URL в строку Адрес и нажмите кнопку OK. Замечательно то, что если некий пользователь обновит календарь Fedora, он будет синхронизирован с вашим, так что у вас всегда будет свежая информация. Выше главного окна календаря расположен список встреч: это просто список всего, что вам предстоит, и поскольку он отражает содержание календаря ниже, то вы можете включать и выключать его части, используя флажки в списке календарей слева.

Что за дела!

Мы заготовили несколько календарей, предоставляющих долгосрочный обзор того, что нас ждет – и можем сконцентрироваться на задачах дня, а для этого необходим Список задач [To Do list]. *Sunbird* имеет окно со списком задач, но бывает удобнее разделить ваше расписание на два вида (долгосрочное и текущее), чтобы, работая над непосредственными задачами, вы не отвлекались бы на то, что ждет вас за углом.

Для управления нашими списками воспользуемся сервисом под названием Remember The Milk [«Не забудь купить молока»] (**www.rememberthemilk.com**). Это сетевое приложение позиционируется как «список напоминаний о покупках», но фактически это Управление задачами/Список задач промышленного уровня, и оно прекрасно подойдет для наших попыток Привести Дела В Порядок. Мы выбрали Remember The Milk (RTM), а не другой сетевой сервис из подобных Списков задач по той причине, что RTM позволяет создавать различные списки, способные отражать пометки о событиях в наших календарях. К тому же последняя версия программы работает с *Google Gears* (более подробно – см. **http://gears.google.com**), а значит, вы сохраните доступ к редактированию ваших списков, будучи вне сети; данные затем будут обновлены при очередном подключении к Интернету.

Прежде всего, надо зарегистрироваться на сервисе и, что не обязательно, загрузить и установить программу *Google Gears*, запускаемую из XPI в *Firefox*. Зарегистрировавшись и войдя, щелкните на ссылке Settings [Настройки], расположенной справа вверху главной страницы RTM. Теперь выберите вкладку Lists [Списки]: здесь вы можете опре-



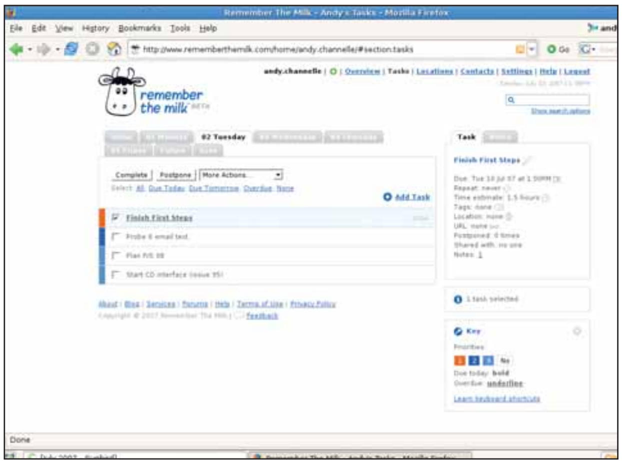
› **Масса календарей на экране может вас запутать. Используйте список календарей слева для уменьшения числа элементов.**

делить сколь угодно много своих списков. Щелкните на кнопке Add List [Добавить список] и дайте списку имя. При желании быть действительно суперсобранным, можете создать список для каждого дня недели, чтобы знать, что вы хотели выполнить в каждый 24-часовой период, плюс список Будущее для того, что произойдет чуть позже. Или просто следуйте схеме, используемой для вашего календаря. Каким бы способом вы ни решили организовать ваши списки, RTM разместит их в алфавитном порядке, а если вы хотите иного упорядочения, придется добавить к вашим спискам числа (01, 02, 03 и т.д.). Покончив с этим, перейдите на страницу Tasks [Задачи] (ссылка справа вверху) – вы увидите, что каждый список имеет свою вкладку; есть еще две стандартные вкладки с именами Inbox [Входящие] и Sent [Отправленные], к которым мы сейчас перейдем.

Чтобы начать, щелкните на одной из вкладок. Метод добавления задачи тот же, что и при создании различных списков: выберите Add Task [Добавить задачу], введите имя задачи и нажмите ввод. По умолчанию, задачи будут появляться в порядке их введения, но вы можете изменить этот порядок, установив некоторым из них приоритеты – это один из фундаментальных аспектов серьезной организации, достойный применения. RTM имеет четыре настройки приоритетов: 1, 2, 3 и none [нет]. Для придания задаче приоритета, выберите ее, используя флажок слева от имени задачи, а затем в выпадающем списке над задачами укажите соответствующий приоритет. Задача при этом подсветится и, в зависимости от выбранных вами настроек, может также изменить свою позицию в списке. Высокоприоритетные элементы всплывут к верху краю списка, элементы без приоритета останутся внизу. Вы можете также повысить или понизить приоритет области задач, выбрав все те, что нужно изменить, и применив к ним пункт списка Move Priority Up... [Повысить приоритет...]. При этом приоритет 3 у задач изменится на 2, а 2 изменится на 1. События без приоритета автоматически получат приоритет 3.

Это облегчает управление приоритетами в течение дня. Завершив задачу с высоким приоритетом, вы можете выбрать ее и пометить как Complete [Выполнено], а затем выбрать следующую и увеличить ее приоритет на 1, чтобы знать новую первоочередную задачу.

Вы могли заметить, что задачи по умолчанию не связаны с датой; добавить ее и можно, и несложно. Сперва выберите задачу, установив флажок, а затем перейдите в раздел справа от списка задач. На вид это просто информационный список, но все записи в нем можно редактировать, и если вы щелкнете на тексте рядом с записью Due: [Срок], то сможете добавить информацию о сроке выполнения задачи. RTM хватает ума, чтобы проверить и реализовать ваши намерения; например, если вы введете срок как 'next tuesday' [следующий вторник], он автоматически введет за вас дату, а если вы укажете только время, он сочтет, что вы имели в виду «сегодня», если время еще не прошло; иначе он впишет его на «завтра». Вы можете также установить опцию повторения (она работает так же, как и в календаре), добавить предположительное время завершения, поместить тэги для категорий поверх уже имеющихся и даже добавить URL, если, например, задача связана с неким web-сайтом или документом в сети.



› Разбейте ваши списки на управляемые куски, используя вкладки RTM; благодаря Google Gears, это можно делать и вне сети.

КПК-органайзер Graphite

Все еще существуют такие места и времена, когда у вас нет доступа к вашим любовно построенным календарям и спискам задач. Для таких случаев, органайзер *Graphite* – прекрасный инструмент внесения данных, готовых для синхронизации с вашими календарями и списками задач при первой возможности. Такие продаются в любом приличном магазине канцтоваров и в большинстве супермаркетов, причем во всевозможных видах: прошитые, на спиральке и клееные. На пару с маленьким карандашом, органайзер можно легко извлечь и использовать практически везде, даже там, где суеверные туземцы подозревают, что электричество – это мстительный злой дух. Основные принципы обращения с подобными штуками – выбирать их малого формата, использовать по листку на день и, вычеркнув все на листке, тут же его выбрасывать, чтобы верхним всегда был «сегодня» или «просрочено». Да, и не забывайте о взаимном обмене событиями между органайзером и вашим списком задач (исключительно нудное занятие).

Скорая помощь

Если какое-то дело требует меньше двух минут, сделайте его прямо сейчас.

В добавление ко вкладке Task [Задача], имеется также опция Notes [Заметки] – это небольшая, автоматически сохраняющаяся текстовая область, где можно внести дополнительные детали конкретной задачи. Все эти детали окажутся под рукой, когда вы выберете задачу из – весьма мощное подспорье. Задачи можно также добавлять в список по электронной почте, SMS или, если вы хотите приобщиться к Web 2.0, даже через учетную запись Twitter.

Полный цикл

Итак, мы организовали наше расписание в *Sunbird* и справились с организацией управления текущими задачами посредством Remember The Milk. Настало время небольшого объединения мыслей и связей: соединим обе службы вместе. RTM имеет возможность вывода вашего списка задач – по крайней мере, тех, у которых есть дата завершения – в виде iCal-файла, то есть мы можем подписаться на этот URL при помощи *Sunbird* и получить автоматическое появление задач в календаре. Стандартная система RTM выводит каждый список в виде отдельного iCal-файла, но не худо бы собрать их в один синхронизированный файл. Так что вновь перейдите на страницу Settings [Настройки] и выберите вкладку Info. Здесь вы найдете адрес электронной почты для пересылки задач в ваш список, адрес ленты Atom (и сможете просматривать ваш список задач в читалке RSS), а также два адреса, которые можно добавить в строку Адрес в окне нового календаря *Sunbird*.

Здесь есть две опции. Во-первых, возможен экспорт списка задач в виде списка, но он поддерживается немногими программами. Лучший (да и единственный) выбор для *Sunbird* – заставить RTM экспортировать списки как события. Щелкните правой кнопкой на URL с пометкой iCalendar Events Service (All Lists) [Служба событий iCalendar (Все списки)] и выберите Copy Link Location [Копировать ссылку]. Теперь вернитесь в *Sunbird* и вставьте эту информацию в строку Адрес нового календаря. Задайте новому календарю имя и цвет, и ваши события перекочуют в *Sunbird*. Однако помните, что в отличие от календаря Google, к которому мы подключались ранее, это езда в один конец: изменение даты завершения задачи в *Sunbird* не отразится в списке RTM.

Хотя хорошее управление временем не снизит число распечаток и не укоротит бессмысленные споры ваших детей, оно может оказать положительное влияние на ваш уровень стресса и улучшить отношения с семьей и коллегами. И благодаря осведомленности о том, что мне следует делать и когда, я выкроил 20 минут на эффектное завершение статьи. А лучше посижу немного в саду... LXF

Bash: Используем

Часть 2 Хотите проводить в командной строке больше времени? **Рэйчел Проберт** расскажет о *grep*, *pipe* и других командах, которые обязательно надо знать.



Наш эксперт

Рэйчел Проберт последние семь лет консультирует начинающих бизнесменов и разрабатывает для них web-приложения.

В прошлый раз мы говорили о действиях, потенциально способных отобрать много времени: например, необходимо быстро найти отчет, написанный когда-то давно. Вы забыли имя файла, но помните, что темой отчета было соглашение с компанией ABC. После обнаружения документа его содержимое нужно отправить коллеге или шефу.

```
grep -r -i -H CompanyABC /home/rjprobert/work
cat filename | mail colleague@work.com
```

Для начала мы разомнемся с помощью нескольких команд для опытных пользователей. Они вряд ли пригодятся обычным пользователям для каждодневной работы, но если вы администратор системы и устанавливаете программы для своих пользователей, значение этих команд повышается.

Команда whoami не вовлечет вас в философские дискуссии по самопознанию, но напомнит, под каким именем пользователя вы вошли в систему; очень удобно, когда вы работаете с несколькими учетными записями, используемыми для решения различных задач, и постоянно переключаетесь между ними. Похожая команда, whatis, позволит понять, что делает та или иная программа. В Linux сотни программ, и невозможно запомнить, что делает каждая из них. Например, напечатав whatis grep, вы увидите:

```
grep (1) - print lines matching a pattern
```

Команда whereis поможет найти другие нужные вам команды или программы. Если вы загружаете из Интернета программу, для работы которой нужна другая программа, вам захочется узнать, установлена ли она в вашей системе. Чтобы выяснить местонахождение программы и ее man-страницы, наберите whereis [имя программы]. Так, например, находится расположение популярного текстового редактора *vi*:

```
whereis vi
```

Echo – небольшая команда, повторяющая все, что вы ни напечатали. Например, если набрать echo hello, на экране появится слово "hello". Вообще говоря, обычно это делать незачем: команда echo может пригодиться вам в будущем, если вы начнете писать скрипты. (Скрипт – это небольшая программа, позволяющая выполнить несколько команд за один раз.) В скриптах echo может использоваться для отображения действий скрипта в каждый момент времени, или чтобы попросить пользователя что-нибудь сделать, например, ввести текст.

Команда last выведет список пользователей, заходивших сегодня в систему, и терминалы, которые они используют (использовали). Набрав exit, вы завершите сеанс работы с терминалом. Если нужно зайти в систему под другим пользователем, используйте вместо нее команду logout.

Часть 1: По Grep'у!

Вот теперь повеселимся! Grep – команда, которая находит слова в файлах. Охотно сказала бы, что ее название произошло от слова «поиск» по-клингонски, но это было бы неправдой и только укрепило бы стереотип, что все технари любят Star Trek! На самом деле название команды – сокращение от «Глобальный Парсер Регулярных Выражений» (Global Regular Expression Parser), что означает «глобальный поиск строк, соответствующих регулярному выражению, и их вывод на экран». Название команды происходит от команды текстового редактора ed g/re/p, выполняющей подобные действия.

Начнем с создания специального каталога, где будем размещать новые файлы. Чтобы убедиться, что каталог создан, используйте команду ls.

```
probert@ubuntu:~$ mkdir Work
probert@ubuntu:~$ ls
Bin Desktop Examples LAN Music Work
```

Откроем текстовый редактор и создадим новый файл **report1.txt**:

```
probert@ubuntu:~$ pico report1.txt
```

Pico – это текстовый редактор, который я использовала при написании этой статьи. В дистрибутив Linux включено несколько текстовых редакторов; выбор одного из них определяется лишь вашими предпочтениями и наличием у редактора специальных функций, которые помогут в решении именно вашей задачи. Когда текстовый редактор откроется, напечатайте слово **CompanyABC**. Затем точно так же создайте второй файл с именем **report2.txt**, который будет содержать слово **companyabc**. Закончив, введите команду ls, и вы увидите примерно такой список:

```
probert@ubuntu:~$ ls
Bin Desktop Examples LAN Music report1.txt report2.txt Work
```

Обратите внимание, что файлы, которые мы создали, находятся рядом с каталогом **Work**. Давайте переместим их в нужный каталог:

```
probert@ubuntu:~$ mv report*.txt /home/probert/Work
probert@ubuntu:~$ ls
Bin Desktop Examples LAN Music Work
```

Звездочка (*) – это специальный символ, который позволяет нам делать все, что мы хотим, с файлами, название которых начинается со слова **report**; каждый из таких файлов включается в команду. У меня есть правила именования файлов и разделения их на категории, которых я строго придерживаюсь. Например, названия всех файлов отчетов начинаются со слова **report**, счетов – со слова **invoice** и т.д. Пускай это выглядит как ритуальные действия безнадежно больного ОКР, зато при наличии методичного подхода нужные вещи находятся очень быстро!

Переместимся в каталог **Work** и снова используем ls, чтобы убедиться, что все файлы перемещены:

```
probert@ubuntu:~$ cd Work
probert@ubuntu:~/Work$ ls
report1.txt report2.txt
```

Сейчас мы сделаем следующий шаг в поисках этого крайне важного файла для нашего коллеги. Команда grep обладает множеством опций (их список выведет команда man grep). Опция -r позволяет выполнить рекурсивный поиск файлов, начиная с названного каталога, а опция -i выполнит поиск без учета регистра. Попробуйте следующие команды с опцией -i и без -i и посмотрите, как это повлияет на результаты поиска.

```
probert@ubuntu:~/Work$ grep -r -i 'CompanyABC' /home/
probert/Work
```

Результат будет примерно таким:

```
/home/probert/Work/report1.txt:CompanyABC
/home/probert/Work/report2.txt:companyabc
```

Итак, мы нашли этот неуловимый файл; что же дальше?

> **Скорая помощь**
>
> Grep требует немалого времени! Если у вас есть хоть какие-то мысли о том, где находится файл, всегда указывайте каталог. Просмотр всех каталогов подряд может ох как затянуться.

КОМАНДНУЮ СТРОКУ

Часть 2: Стандартный ввод-вывод

Стандартный ввод/вывод позволяет вводить данные в программу из любого источника и выводить данные в любой источник. Таким образом, можно направить вывод одной команды непосредственно на ввод другой команды... правда, мило? Linux предоставляет три канала ввода/вывода, более известных как потоки ввода/вывода. По умолчанию стандартный ввод подразумевает клавиатуру, стандартный вывод – экран. Третий поток ввода/вывода – это на самом деле поток вывода для сообщений об ошибках, предупреждений и оповещений: так легче отличить сообщения об ошибках от остальных сообщений. Поток ошибок обычно также выводится на экран, но (как и каждый из трех стандартных потоков) может быть перенаправлен. Этим мы сейчас и займемся.

Есть два основных способа перенаправления потоков данных; мы можем использовать > или | (перенаправление и каналы соответственно). Если нужно просто направить стандартный вывод в файл, можно напечатать команда > файл. Давайте попробуем это на реальном примере: найдем все файлы с именем passwd в каталоге **/etc**. Наберите

```
find /etc -name passwd
```

Результат будет примерно таким (он зависит от особенностей вашего дистрибутива Linux):

```
probert@ubuntu:~$ find /etc -name passwd
find: /etc/ssl/private: Permission denied
/etc/passwd
/etc/pam.d/passwd
find: /etc/cups/ssl: Permission denied
```

Обратите внимание, что команда генерирует как стандартный поток вывода, так и поток ошибок. Сейчас мы создадим файл **findresult** и перенаправим стандартный поток вывода в этот файл, оставив на экране только ошибки.

```
probert@ubuntu:~$ find /etc -name passwd > findresult
find: /etc/ssl/private: Permission denied
find: /etc/cups/ssl: Permission denied
```

Перенаправим эти ошибки в другой файл, **finderrors**, используя 2>.

```
probert@ubuntu:~$ find /etc -name passwd 2> finderrors
/etc/passwd
/etc/pam.d/passwd
```

Наберите ls, и вы получите результат, похожий на мой.

```
probert@ubuntu:~$ ls
Bin Desktop Examples finderrors findresult LAN Music
```

Чтобы прочитать содержимое этих файлов, можно открыть их в текстовом редакторе pico:

```
probert@ubuntu:~$ pico findresult
```

Я предпочитаю просматривать небольшие файлы с помощью команды cat. Ниже я использовала эту команду для просмотра результатов поиска в файлах **findresult** и **finderrors**.

```
probert@ubuntu:~$ cat findresult
/etc/passwd
/etc/pam.d/passwd
probert@ubuntu:~$ cat finderrors
find: /etc/ssl/private: Permission denied
find: /etc/cups/ssl: Permission denied
```

Вы также можете добавить информацию в файл с помощью команды >>. Давайте перенаправим стандартный вывод, чтобы добавить информацию в наш первый файл **findresult** следующим образом:

```
probert@ubuntu:~$ find /etc -name passwd 2>> findresult
```

Вы можете снова проверить результаты с помощью команды cat findresult. Второй способ перенаправления потоков – вывод одной команды прямо на вход другой (канализация). Давайте проверим содержимое каталога **/usr/lib**, где находятся объектные файлы, библиотеки и системные двоичные файлы. В моем каталоге **/usr/lib** примерно полторы тысячи файлов – многовато, чтобы прокручивать этот список один за другим! Если мы перенаправим вывод ls на ввод команды less, то получим список, в котором можно перемещаться по страницам, в удобном для нас темпе. Попробуйте:

```
ls /usr/lib | less
```

Достигнув конца документа, нажмите клавишу Q, чтобы вернуться в командную строку. Вы также можете предварительно отсортировать этот список по убыванию, например, таким образом:

```
ls /usr/lib | sort -r | less
```

Краеугольный камень философии Unix заключается в том, чтобы каждая программа делала что-то одно по-настоящему хорошо, позволяя использовать свой вывод как ввод другой программы. Использование каналов позволяет провести этот принцип в жизнь, создавая очень эффективные цепочки команд.

Теперь, разузнав, как работают *grep* и каналы, сведем их вместе для поиска того самого отчета и отправим его нашему коллеге:

```
grep -r -i -H CompanyABC /home/rjprobert/work
cat report1.txt | mail colleague@work.com
```

Миссия завершена! LXF

Помощь по grep

Поскольку *grep* производит поиск соответствий по шаблонам, вы должны предоставить ему регулярное выражение для этого шаблона. Регулярные выражения бывают простыми и сложными. Наиболее распространенные (и простые) типы регулярных выражений выглядят так:

» **abc** Соответствует строкам, которые содержат "abc" в любом месте.

» **^abc** Соответствует строкам, начинающимся с "abc".

Скорая помощь

Не используете *grep* с регулярными выражениями? Вот основные опции:

» **-i** Игнорировать регистр при сравнении.

» **-v** Выводить только строки, не соответствующие шаблону.

» **-c** Вывести количество файлов, соответствующих шаблону.

» **-n** Отображать номера строк, где найден шаблон.

» **-H** Выводить имя файла для каждого соответствия.

Мини-проект: команды для умных

Есть несколько общих команд, о которых полезно знать. Привожу некоторые из них без объяснений; эти команды можно использовать независимо от дополнительных опций. Наберите их и посмотрите на результат.

date	df
date +%D	free
date +%T	du
cal	top
cal 1066	ps

» **Через месяц** Монтирование, создание резервных копий, распаковка файлов из tar и zip.

Управляемся:

Надоело мириться с дезориентирующими тэгами и нелепыми именами файлов? Укодируйте их до полного исчезновения с помощью **Ника Вейча** и магии Python!





Наш эксперт

Ник Вейч
Добродушный олигарх Linux Format очень организован – в электронном смысле. Но бардак в его сарае такой же, как и в вашем: велосипеды, перевязанные пожарным шлангом, кучи старых банок из-под краски…

Правда-правда, я ужасно люблю порядок: все вещи на своем месте, все под рукой, и вы можете мигом определить, что есть, чего нет; и всякое такое. Впрочем, люди, видевшие фото моего рабочего стола или совавшие нос в мой бельевой ящик, пожалуй, оторопеют от явной лживости этого высказывания. Но, пожалуйста, внимательно прочтите его еще раз: я сказал, что люблю порядок, а вовсе не что я люблю его наводить. Моя мечта – чтобы во всех комнатах моего дома стояло по хитроумной мусорной корзине, связанной с этаким интеллектуальным сканером объектов: каждый раз, покончив с использованием/чтением/изучением чего-либо, я бы просто скидывал эту штуку в мусоросборник, а она волшебным образом перемещалась куда-нибудь и дожидалась там своего часа. Увы, похоже, подобные технологии не скоро до меня доберутся. Сейчас самая моя больная тема – музыкальная коллекция. Я, конечно, целиком за свободу творчества, но то, что все мои сборники CD имеют разные шрифты, цвета и формат хранения данных, действует на нервы. Впрочем, виртуальный мир вашей Linux-системы способен удовлетворить любые ваши прихоти, включая страсть к педантизму и скрупулезной правильности!

Что такое «кортеж»?

Кортеж – это объект, состоящий из нескольких значений. Он часто используется в Python. Хороший пример использования кортежа – цвет. Вместо того, чтобы хранить три значения в трех разных переменных для красного, зеленого и синего цветов, вы назначаете одну переменную, содержащую три значения! Кортеж в Python записывается примерно так: (123,255,17). Вы можете получить доступ к любому элементу кортежа с помощью индексов – значений в квадратных скобках, указываемых после имени переменной. Например, команда print Colour[1] вернет значение «255» в случае с предыдущим примером – только не забывайте, что значения индексов начинаются с 0!

Python – превосходный язык для выполнения стандартных операций с файлами. Он позволяет обрабатывать имена файлов и использовать функциональные возможности ОС с помощью ряда встроенных функций, всегда готов при необходимости вызвать внешние утилиты, имеет огромную коллекцию готовых модулей, способных выполнить практически любые действия, и к тому же очень прост для понимания. Он хорош не только для прототипирования, но и для создания самостоятельных приложений – или как минимум удобных небольших сценариев для выполнения частных задач.

Итак, наша цель – разобраться с коллекцией музыкальных файлов на моем Linux-компьютере (отдельным счастливчикам это тоже пригодится). У меня есть один большой каталог под названием 'Music', который, как полагается, делится на подкаталоги по исполнителям, альбомам и отдельным трекам. Однако, несмотря на столь стройную систему, у меня имеются проблемы непосредственно с музыкальными файлами. Вот примерный список вещей, мешающих мне насладиться моей коллекцией:

1. Некоторые файлы имеют слишком длинные имена, затрудняющие работу с ними. Такое бывает с покупной музыкой: в название включается сразу и альбом, и имя исполнителя, и номер дорожки, и т.д. Я же хочу единый для всех названий вид, желательно без дурацких символов.
2. Некоторые файлы содержат тэги формата ID3v1, а не ID3v2. Беда небольшая, но по возможности я бы предпочел наслаждаться и преимуществами ID3v2.
3. Некоторые файлы снабжены тэгами формата ID3v2, но не ID3v1. Из-за этого они не читаются на стареньком MP3-плейере в моем автомобиле.
4. Некоторые из файлов неработоспособны. Неплохо было бы иметь список дефектных файлов, чтобы удалить их из коллекции или заменить.
5. Некоторые из файлов сохранены в нетипичных форматах, которые распознаются не всеми доступными устройствами. Список этих файлов тоже пригодился бы.

Но как же достичь хотя бы части этих целей? Хоть я и не вполне уверен, что все перечисленные мной желания легко выполнимы, но попытаться стоит; по крайней мере, я знаю, с чего начать. В Python'е имеется модуль под названием *os*. Он реализует все стандартные функции ОС, в частности, связанные с файловой системой. Python – кросс-платформенная разработка, поэтому, несмотря на различия реализации в разных операционных системах, задача у данного модуля одна: предоставление функциональности уровня ОС с единым интерфейсом. Одной из самых полезных из известных мне функций, представленных в данном модуле, является walk(). Если задать ей имя родительского каталога, эта функция вернет список кортежей (см. врезку слева), включающих имя родительского каталога, список подкаталогов (если они есть) и список имен файлов. Таким образом, чтобы получить список всех файлов, включая полный путь к ним, нужно выполнить функцию walk() с заданным нами каталогом и по шагам обработать результат.

Python и музон

```
import os
topdir="/usr/share/music/"
walklist = os.walk(topdir)
for dirs in walklist:
  if(dirs[2]):
    for file in dirs[2]:
      // Выполнить какое-нибудь действие
      print os.path.join(dirs[0],file)
  else:
    for entry in dirs[1]:
      print dirs[1]
```

В данном примере мы сначала импортируем модуль os, затем назначаем каталог, который хотим использовать. В результате выполнения функции walk() генерируется объект в переменной walklist. С помощью конструктора for мы можем затем использовать имеющийся в Python способ пошагового перемещения по списку для работы с каждым элементом. Чтобы вам стало понятнее, данные переменной walklist должны выглядеть примерно так:

```
('/usr/share/music/Blonde_Redhead', ['1980 Forward', '23'], [])
('/usr/share/music/Blonde_RedHead/1980 Forward', [], ['18-Magic
Mountain.mp3'])
('/usr/share/music/Blonde_Redhead/23', [], ['10-My Impure Hair.
mp3','1-23.mp3', '7-Publisher.mp3', '3-The Dress.mp3', '6-Silently.
mp3'])
```

Теперь пошагово проверим каждую строку. Если элемент строки является файлом (одним или несколькими, в виде списка), мы хотим выполнить с ним определенные действия. Оператор if проверяет, имеется ли что-нибудь в каталоге, а если имеется, мы продолжим перемещаться по списку файлов тем же способом. Что делать с файлами, пока не совсем ясно, поэтому просто распечатаем список в стандартный вывод. Здесь мы воспользуемся другой полезной функцией модуля *os* – os.path.join(). Она соединяет вместе все компоненты пути к файлу и записывает их в виде, соответствующем текущему модулю os, что, несомненно, гораздо лучше, чем простое соединение вхождений с помощью слэша /, как это обычно делается в Linux!

Я также добавил здесь еще одну ветку else – на случай, если мы захотим что-то проделать и с каталогами (может, переименовать?). Но для начала просто выполните данный скрипт, чтобы проверить, работает ли он в вашей системе. Главное – не забудьте указать именно тот каталог, где действительно имеются MP3-файлы.

Проверка типов файлов

Ну вот, теперь самое время добавить новые функции в данный скрипт. Но прежде чем приняться за файл, я предлагаю убедиться, что он действительно музыкальный! Думаю, мы можем смело предположить, что все MP3-файлы имеют расширение mp3. Таким образом, в нашем цикле мы могли бы проверить это и пометить файлы, не соответствующие данному критерию. Вероятно, мы могли бы просто включить такую проверку внутри цикла, уже осуществляемого оператором if, но так как проверка может потребоваться не одна, гораздо разумнее присвоить расширение переменной и затем проверять ее. Для извлечения расширения воспользуемся простым и мощным способом обработки строк и переменных других типов в Python (подробнее – см. врезку справа вверху). Добыв расширение файла, можно проверить, не совпадает ли оно с другими форматами музыкальных файлов, вроде '.wma' и '.ogg' – на случай, если мы захотим поработать и с такими файлами. В Python нет оператора case:; придется нагромоздить последовательность операторов if, elif (else if) и else. Символ # в Python указывает, что оставшаяся часть строки – комментарий. В итоге наш основной цикл примет такой вид:

```
for file in dirs[2]:
    fullpath = os.path.join(dirs[0],file)
    extension = file[-4:]
    if (extension!='.mp3'):
               # Это не MP3-файл; проверить на другие расширения?
      if (extension =='.wma'):
          wmalist.append(fullpath)
      elif (extension =='.ogg'):
          ogglist.append(fullpath)
      else:
          unknownlist.append(fullpath)
    else:
      # Это MP3-файл.
      # Что-нибудь с ним сделаем.
      print fullpath
```

Мы также должны объявить три переменных типа «список» в основном коде. Возможно, вы захотите добавить код, обрабатывающий файлы, или указать другие варианты расширений. Для полноты картины, создадим также список неизвестных типов файлов, найденных в нашем каталоге – чисто информативный; заодно он поможет обнаружить то, что мы упустили.

Теперь займемся MP3-файлами. Спецификация ID3v1 весьма незамысловата: в конец файла просто добавляется 128 байт информации о файле, и легко создать синтаксический анализатор для сбора этих данных и дальнейшей обработки. Однако ID3v2 придерживается других стандартов: информация заносится в начало файла, в виде сложной структуры с переменной длиной данных, и, если честно, нам пришлось бы изрядно разрастить наш маленький скрипт, чтобы с ней справиться. Поэтому мы поступим, как все нормальные программисты: схитрим! Существует множество готовых модулей Python, способных читать данные MP3-тэгов, вот и возьмем один из них. Рекомендую *EyeD3* – это признанный фаворит, и в нем есть все, что нам нужно. Вы найдете его на DVD, прилагаемом к данному журналу, а на сайте http://eyed3.nicfit.net может оказаться более свежая версия.

Данный модуль творит всяческие чудеса; некоторые из них нам даже и трогать незачем. Вы можете почитать о нем подробнее на ука- »

Будем резать!

В Python используются индексы для «нарезания на кусочки» (иногда говорят «слайсинг», от англ. slicing) строк, списков и других типов переменных. Пусть у нас имеется запись plop.mp3; мы можем запросто извлечь из нее любой нужный нам бит. Откройте терминал, запустите Python для входа в интерактивный режим и попробуйте следующие команды:

```
>>>string = 'plop.mp3'
>>>print string[1]
l
>>> print string[:2]
pl
>>> print string [-0]
p
>>> print string [-1]
3
>>> print string [-4:]
.mp3
>>>
```

В последнем операторе print из строки выхватываются четыре последних символа. Отрицательные индексы отсчитываются с конца строки, но будьте внимательны: -0 означает то же самое, что и просто 0, а именно – начало строки! (Индексы в Python нумеруются с 0.)

Скорая помощь

Тэги ID3v2 могут содержать массу избыточных данных: например, битрейт, лицензионную информацию или даже текст песни. Если у вас портативный плейер с небольшим объемом памяти, ликвидация этого балласта может значительно увеличить

» занном сайте, но для наших целей достаточно импортировать модуль и ознакомиться с функциями, которые нам пригодятся.

К счастью, модуль eyeD3 не относится к числу сложных. Опасения внушает только один объект – eyeD3.Tag, содержащий структуру тэгов ID3v1 и ID3v2. Чтобы заполнить его данными из вашего файла, вы должны использовать метод объекта link с соответствующим именем файла, и выглядит это примерно так:

```
mytag=eyeD3.Tag()
mytag.link('/usr/share/music/Blonde_Redhead/23/1-23.mp3')
print mytag.getArtist()
print mytag.getAlbum()
print mytag.getTrackNum()
```

Изменив любой из тэгов, просто вызовите метод tag.update(), и он запишет новый тэг в файл. Теперь разберемся, как мы поступим с нашими файлами. Вместо того, чтобы сразу создавать необходимый код, я обычно просто вписываю набор операций, которые нужно сделать, в виде комментариев. Потом можно легко сортировать их, разбивать на более мелкие действия, или, в конце концов, собраться с духом и дописать требуемый код! Ниже представлен примерный список операций над нашими файлами, которые могли бы понадобиться:

- Получить расширение файла
- Он заканчивается на .mp3?
- Проверить, действительно ли это MP3-файл
- Прочитать тэги версии 2
- Прочитать тэги версии 1
- Если имеется только один тип, копировать в другой тип
- Если тэгов вообще нет, установить пометку
- Создать нормальное имя файла из тэгов
- Предложить/записать новое имя файла
- Проверить, соответствует ли название альбома имени каталога
- Если это не MP3-файл, то что? Добавить в соответствующий список

Как сказано выше, впишем это в скрипт в виде комментариев. Обычно приступать к делу лучше с первого пункта, чтобы от вас не ускользнули логика действий и структура скрипта. Особо хитрые участки – или участки, используемые многократно – можно вынести в отдельную функцию. Существуют мнения и за и против этого, и программисты нередко обсуждают, что допустимо выделять в отдельную функцию. Теория выглядит примерно так: чем больше вы выносите из основного цикла программы, тем более очевидной становится общая задача. Но есть и контрдовод: чем больше вы включаете подобных обобщений, делая программу максимально простой и понятной, тем сложнее потом добраться до работы отдельных функций. Резюме: поскольку вы пишете программу исключительно для личного пользования, делайте так, как вам удобнее!

Проверка типов файлов – 2

Теперь пора и делом заняться. Прежде всего, получим расширения файлов через слайсинг Python'а, а затем проверим, MP3 это или нет. После чего употребим модуль *eyeD3*, чтобы распознать, действительно ли файл соответствует заявленному формату.

После операторов проверки сделаем отступы, чтобы было видно, какой блок кода используется при обнаружении соответствия, и возвратимся к оператору else. В данном случае я слегка поменил порядок действий, для начала разобравшись с файлами, не соответствующими формату MP3, а затем через оператор else перешел к дальнейшим действиям. Почему? Да просто очень легко позабыть про файлы, не интересующие вас в данный момент; если вы можете парой строк отделаться от второстепенных данных, лучше сразу с ними и покончить. В конечном итоге, быть может, у нас появится специальная функция для их обработки.

В представленном ниже коде можно встретить так называемые функции «санации» (sanitize). Присутствие их тела в основном коде сделало бы его неудобочитаемым, кроме того, иногда их бывает нужно вызывать из разных мест. Наконец, так с их реализацией можно разобраться и позже, не затрагивая основной код.

```
if (extension !='.mp3'):
        # Не файл MP3? Проверить на другие расширения?
    if (extension == '.wma'):
        wmalist.append(fullpath)
        # logger.warn("silly wma file %s",fullpath)
    elif (extension == '.ogg'):
        ogglist.append(fullpath)
    else:
        unknownlist.append(fullpath)
else:
    # Это вправду MP3-файл?
    if eyeD3.isMp3File(fullpath):
        # Прочитать тэг ID3V2
        tag2 = eyeD3.Tag()
        tag1 = eyeD3.Tag()
        a = tag2.link(fullpath,eyeD3.ID3_V2)
        b = tag1.link(fullpath,eyeD3.ID3_V1)
        if b and not a:
            # Имеется только тэг ID3v1
```

Скорая помощь

Тэги ID3v1 могут вызывать в динамиках очень короткие щелчки между дорожками на старых MP3-плеерах. Эта проблема не решается путём конвертирования тэгов в ID3v2, т.к. многие устройства не поддерживают новый формат. Музыку прослушивать вы сможете, но на дисплее не будет отображаться информация о дорожках.

Форматирование

Необходимость чёткого форматирования кода в Python многих ставит в тупик, но это вполне простое и разумное требование. В частности, отступы строк (посредством пробелов или табуляции) в блоке кода должны быть одинаковыми, потому что в Python отступы являются значимыми. Взглянув на код, вы не увидите фигурных скобок вокруг любых операторов или блоков кода: Python распознаёт окончание блока по окончанию отступа. Это даёт сразу два преимущества: код становится гораздо удобнее для чтения, и не надо заботиться о вложенных фигурных скобках!

Журналирование

Небольшие скрипты часто нашпиговывают множеством выводов промежуточных данных на печать, чтобы знать, что и как происходит. Это действительно удобно; но иногда хочется превратить строку с оператором печати в комментарий, потом снова её подключить, и т.д. Проще всего в таком случае использовать встроенный в Python модуль ведения журнала! Он умеет выводить сообщения и в консоль, но способен различать сообщения по значимости, и вы сможете гибко регулировать количество выводов на консоль путём правки всего одной строки кода скрипта. Добавьте этот модуль к списку импортируемых и впишите следующий код в начало вашего скрипта:

```
logger = logging.getLogger("pymusic")
# Создать обработчик консоли
handler = logging.StreamHandler()
# Создать форматирование и настроить обработчик на его использование
formatter = logging.Formatter("%(asctime)s - %(name)s - %(levelname)s - %(message)s")
handler.setFormatter(formatter)
# Добавить обработчик к модулю ведения журнала
logger.addHandler(handler)
logger.setLevel(logging.WARN)
```

Часть этих штук смахивает на волшебство, и кое-кто считает модуль ведения журнала непостижимым, но на самом деле все очень просто. Если вы хотите узнать больше, документация находится здесь: http://docs.python.org/lib/module-logging.html. Фактически, всё, что необходимо знать – это набор стандартных команд для управления выводом сообщений в консоль. Например, следующий код:

```
logger.warn(" file %s does not seem to be a valid mp3 file",fullpath)
```

выведет в консоль только сообщения, относящиеся к предупреждениям типа WARN или ниже (порядок такой: Critical, Error, Warning, Info, Debug). Можно установить и собственные уровни, но мы займёмся этим в другой раз...

```
            print "version1 only"
            # Создать информацию для tag2 по tag1
            print fullpath
            artist = tag1.getArtist()
            album = tag1.getAlbum()
            title = tag1.getTitle()
            print artist,album,title
            tag1.update(eyeD3.ID3_V2)
            # check tags are cool
        elif a and not b:
            # Имеется только тэг ID3v2
            print fullpath
            artist = tag2.getArtist()
            album = tag2.getAlbum()
            title = tag2.getTitle()
            print artist,album,title
            try:
                tag2.update(eyeD3.ID3_V1_1)
            except UnicodeEncodeError:
                logger.error(“tag invalid for v1.1 in file %s”,
fullpath)
        elif a and b:
            # Имеются оба тэга
            logger.info( “both versions fine %s”, fullpath)
        else:
            # Тэгов нет вообще
            logger.warn(‘this file has no tags! %s’, fullpath)
            error_flag = True
            # М.б. удастся вытянуть что-нибудь из имени
каталога, где сидит файл!
        if not error_flag:
            # Пускай имя файла будет
            #  number-name.mp3
            title=tag2.getTitle()
            title = title.replace(‘ ‘,’_’)
            n=tag2.getTrackNum()
            # Номер (number) у нас есть
            ns = str(n[0])
            if len(ns)==1:
                ns= ‘0’+ns
        ns=ns+’-’+title+’.mp3’
        # Уберем нехорошие символы
        ns=sanitize(ns)
        if (file!=ns):
            logger.info(“change filename suggested for %s, to
%s!”,filename,ns)
            os.rename(fullpath, os.path.join(dirs[0],ns))
    else:
        # Выдадим предупреждение
        logger.warn(“ file %s does not seem to be a valid mp3
file”,fullpath)
```

Полная распечатка этого небольшого скрипта представлена на DVD данного номера Linux Format. Но будьте осторожны – скрипт не завершен! Нельзя гарантировать, что он не испортит вашу музыкальную коллекцию! Особенно следует обратить внимание на проверку символов в кодировке Unicode, нередко встречающихся в тэгах ID3v2 (недопустимых в тэгах первой версии). Также вы, возможно, захотите обработать и файлы формата **.ogg**.

Здесь не доделано также множество проверок на ошибки. В случае с тэгами в кодировке Unicode поможет структура try: ... except:; она же пригодится и для других случаев (например, выяснении, как поступать с музыкальными файлами без прав доступа на запись). Но я надеюсь, что рассмотренный в статье пример показал вам, как просто создавать полезные скрипты! LXF

Скорая помощь

Хотите разобраться с мелодиями, не имеющими данных об исполнителе или названии альбома? Посетите сайт http://musicbrainz.org, там подскажут.

› Аудиоплейеры вроде *AmaroK* помогают редактировать тэги к каждому файлу, но не показывают явно тип используемых тэгов – ID3v1 или ID3v2.



Сравнение аудиоплейеров в Linux

Приложение	Импорт (секунды)	Использование CPU (%)	MP3	Ogg	FLAC	Загрузка плейлистов	Интернет радио/ потоковые данные	Захват данных с CD	Запись CD	Поддержка MP3-плейеров
Amarok	113	1	✓	✓	✓	✓	✓	✓	✓	✓
Banshee	73	10	✓	✓	✓	☒	✓	✓	✓	✓
BMPx	108	10	✓	✓	✓	✓	Только Shoutcast/Icecast	✓	☒	☒
Juk	56	0.5	✓	✓	✓	✓	☒	☒	☒	☒
Rhythmbox	773	10	✓	✓	✓	✓	✓	✓	✓	только iPod
XMMS	N/A	0.5	✓	✓	Плагин	✓	✓	☒	☒	☒

» **Через месяц** Секреты Python снова помогут вам контролировать события.

Linux: Фильтры

Хватайте трубку и ласты и ныряйте в загадочный подводный мир фильтров и каналов вместе с **Крисом Брауном**.





Наш эксперт

Д-р Крис Браун внештатный консультант по Linux со степенью доктора наук по физике элементарных частиц и обладатель сертификатов Novell CLP и Red Hat RHCE. Он только что написал книгу по SUSE Linux для O'Reilly.

Когда я впервые столкнулся с Unix и Linux, мое внимание сразу привлекла классная вещь: набор средств командной строки – фильтров, и штука под названием «каналы», их соединяющая. Вместе они образовывали невероятно мощную компонентную архитектуру, предназначенную для обработки потоков текстовых данных.

Фильтр – это программа, которая считывает один входной поток, как-то преобразует его и выводит результат в один выходной поток (**Рис. 1**). По умолчанию выходной поток (также называемый стандартным выводом или просто stdout) связан с окном терминала, где запущена программа, а входной поток (стандартный ввод или просто stdin) – с клавиатурой. Однако на практике фильтры редко используются для обработки данных, набираемых на клавиатуре вручную. Если фильтру через аргумент командной строки передать имя файла, он откроет этот файл и считает его содержимое вместо считывания данных со стандартного ввода stdin (**Рис. 2**) – такая схема применяется гораздо чаще. Во врезке напротив показано несколько простых команд. Большинство фильтров сами по себе не делают ничего впечатляющего. Гораздо интереснее использовать их в сочетании друг с другом.

Сочетание фильтров: каналы

Канал позволяет направить данные с выходного потока одного процесса на входной поток другого. Обычно они используются для подключения стандартного потока вывода к стандартному потоку ввода. Создать канал в командной строке очень просто: для этого используется символ | (вертикальная черта). Например, когда в командную оболочку поступает следующая команда:

```
$ sort foo.txt | tr '[A-Z]' '[a-z]',
```

то оболочка одновременно запускает два отдельных процесса для программ *sort* и *tr* и создает канал, передающий стандартный вывод команды sort (процесс-источник) на стандартный ввод команды tr (процесс-получатель). Схема этого процесса показана на Рисунке 3. Приведенная команда tr, если вам интересно, заменяет все символы из набора [A-Z] соответствующими символами из набора [a-z], то есть преобразует текст из верхнего регистра в нижний.

Полуфильтры

Многие команды Linux, которые на самом деле не являются фильтрами, выводят результаты своей работы в стандартный поток вывода stdout и могут стать началом канала. К ним относятся ls, ps, df, du и многие другие. Например, команда

```
$ ps aex | wc
```

пересчитает запущенные на компьютере процессы, а команда

```
$ ls -l | grep '^l'
```

выведет только символические ссылки для текущего каталога. (Регулярное выражение ^l означает строки, начинающиеся с буквы 'l'.)

Менее распространен другой вариант полуфильтров. Это программы, считывающие данные из потока ввода, но не выводящие их в поток вывода stdout. В голову приходят только команда просмотра файлов *less*, утилита печати *lpr* и почтовый клиент для командной строки *mail*. Такие программы могут использоваться как окончание канала; например, команда

```
grep syslog /var/log/messages | less
```

выводит строки файла **/var/log/messages**, относящиеся к системному журналу syslog, с помощью команды *less*. Команда *less* используется как окончание канала очень часто.

Потоковый редактор sed

Потоковый редактор *sed* поддерживает автоматическое редактирование текста и является более гибким по сравнению с большинством фильтров. Он считывает свои входные данные строку за строкой из стандартного потока ввода или из заданного файла, применяет к ним одну или несколько операций редактирования и выводит строки результата в стандартный поток вывода. У этого редактора целый набор команд, но самая полезная – команда замещения. Для начала рассмотрим простой пример использования *sed* для подстановок в тексте, введенном с клавиатуры (stdin):

```
$ sed 's/rich/poor/g'
He wished all men as rich as he
He wished all men as poor as he
And he was as rich as rich could be
And he was as poor as poor could be
^D
$
```

› Рис. 1



› **У фильтра есть один входной поток и один выходной поток.**

и каналы

Полезные фильтры

Фильтр	Что он делает
cat	Копирует входной поток в выходной.
head	Показывает начало файла (по умолчанию – первые 10 строк).
tail	Показывает конец файла (по умолчанию – последние 10 строк).
wc	Подсчитывает символы, слова и строки.
sort	Сортирует входные строки.
grep	Находит строки, соответствующие регулярному выражению.
tr	Преобразует или удаляет символы заданного набора.
sed	Потоковый редактор.
uniq	Оставляет только одну строку из набора одинаковых строк.
awk	Очень гибкая обработка полей данных.

Наверное, нужно немного пояснить. Команда sed заменяет все имеющиеся строки 'rich' строками 'poor'. Суффикс /g говорит о том, что это глобальная замена – если в строке больше одного вхождения, то все они будут заменены (без этого суффикса было бы заменено только первое). Сразу после команды sed мы видим две пары строк. Первая строка в каждой паре – это текст, который мы ввели с клавиатуры, а вторая – результат работы редактора, выведенный в стандартный поток вывода.

А вот более полезный пример, в котором мы используем регулярное выражение для удаления всех полей строк файла **/etc/passwd**, кроме первого:

```
$ sed 's/:.*//' /etc/passwd
root
daemon
bin
sys
... остальные строки пропущены ...
```

Здесь текст, который мы заменяем, определяется регулярным выражением :.*; оно соответствует фрагменту от первого двоеточия до конца строки. Строка, которой будет заменен такой фрагмент (между вторым и третьим слэшами), пуста, что приводит к удалению всех фрагментов, соответствующих регулярному выражению. Так мы получаем список имен пользователей из файла **passwd**. Однако давайте проясним кое-что: отнюдь не меняет файл **/etc/passwd**. Он просто считывает его и выводит измененные строки в стандартный поток вывода.

Awk

Названный в честь своих создателей Ахо [Aho], Вайнбергера [Weinberger] и Кернигана [Kernighan], awk представляет собой отдельную категорию: это развитый язык программирования с переменными, циклами, условиями и функциями. Программа на языке *awk* состоит из одной или нескольких пар «шаблон-действие»:

```
шаблон { действие }
шаблон { действие }
```

Шаблон – это некое условие, применяемое к каждой строке; если строка соответствует шаблону, над ней выполняется указанное действие. Если шаблон опущен, действие применяется ко всем строкам. Если опущено действие, строка целиком выводится на экран. Awk отлично подходит для обработки текстовых данных, разделенных на поля (столбцы); он считывает входные данные строку за строкой и автоматически разбивает их на поля, доступ к которым производится через специальные переменные $1, $2, $3 и т.д.

Для демонстрации работы awk мы будем использовать небольшой набор географических данных, за правильность которых ручается атлас мира Collins Complete World Atlas. Они включают названия стран, их площадь, население (в тысячах), языки и валюту. Для краткости ограничимся четырьмя строками, которые выглядят вот так:

Страна	Площадь	Население	Языки	Валюта
Албания	28748	3130	Албанский,греческий	Лек
Греция	131957	11120	Греческий	Евро
Люксембург	2586	465	Немецкий,французский	Евро
Швейцария	41293	7252	Немецкий,французский,итальянский	Франк

Эти данные находятся в файле с именем **geodata**.

Многие awk-программы так просты, что их можно ввести с командной строки как аргумент awk. Например:

```
$ awk '{ print $1, $5 }' geodata
Страна Валюта
Албания Лек
Греция Евро
Люксембург Евро
Швейцария Франк
```

Эта программа на языке awk содержит единственную пару «шаблон-действие». Шаблон пропущен, поэтому действие применяется к каждой строке – это вывод первого и пятого полей, то есть названия страны и ее валюты.

Теперь давайте найдем страны, которые используют евро. Чтобы »

Скорая помощь

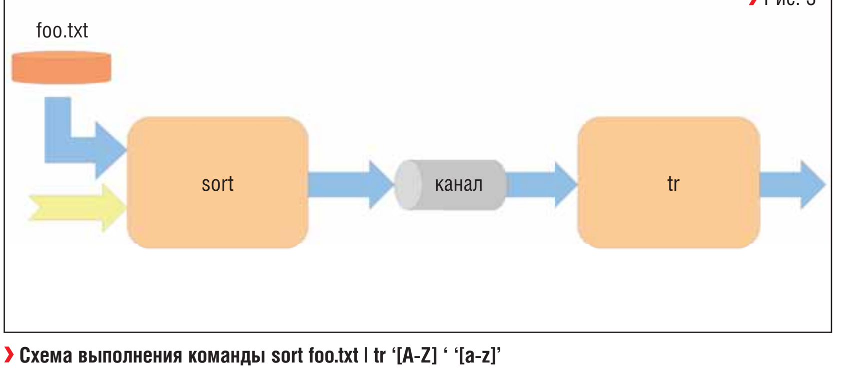
При создании цепочек команд, связанных каналами, лучше делать это по шагам, на каждом этапе проверяя вывод. Можно добавлять команды интерактивно, используя стек истории команд, или записывать их в файл скрипта. Последний подход имеет преимущество сохранности и упрощает повторное использование кода.

Рис. 2



› Получая файл, фильтр читает его, игнорируя стандартный ввод.

Рис. 3



› Схема выполнения команды sort foo.txt | tr '[A-Z] ' '[a-z]'

Скорая помощь

Возьмите в привычку заключать аргументы командной строки в одинарные кавычки, для уверенности в том, что оболочка не распознает их как специальные символы и не выкинет что-нибудь странное. Вреда от них нет, а нужные кавычки помогут избежать разного рода неожиданностей.

» применить условие к пятому полю, можно использовать шаблон, например, таким образом:

```
$ awk ‘ $5==”Евро” { print $1 }’ geodata
Греция
Люксембург
```

А как определить суммарное население всех стран? Это потребует двух пар «шаблон-действие»: одна из них будет срабатывать в каждой строке и последовательно накапливать значения численности населения (из третьего столбца), а вторая сработает только в конце и выведет результат. Можно было бы ввести эту программу через командную строку, как и предыдущие примеры, но на сей раз она немного длиннее, и удобнее записать ее в отдельный файл – я назвал его **totalpop.awk**. Он выглядит так:

```
{ sum += $3 }
END { print sum }
```

У первого действия нет шаблона, поэтому оно применяется к каждой строке. Sum – просто имя созданной мною переменной. В *awk* переменные не нужно объявлять заранее, они начинают существовать после первого упоминания имени (тогда же им присваивается нулевое значение). Второе действие использует специальный шаблон ‘END’, который выводит результат. Он срабатывает только один раз – после того, как все входные данные обработаны.

Теперь я могу запустить awk и заставить интерпретатор считать программу из файла **totalpop.awk**, например, так:

```
$ awk -f totalpop.awk geodata
21967
```

Обратите внимание, что для указания имени файла с программой мы используем ключ -f.

Попробуем найти страны, население которых говорит на указанном языке. Это немного сложнее, так как для каждой страны указан список языков, разделенных запятыми. Нам нужно разделить этот список на составляющие. К счастью, для этого в языке awk есть встроенная функция. Вот полный текст программы, которую я назвал **language.awk**:

```
{ NL = split($4, langs, “,”);
  for (i=1; i<=NL; i++)
    if ( langs[i] == “Греческий”)
      print $1
}
```

Здесь только одно действие, с которым не связан ни один шаблон (поэтому оно применяется к каждой строке), однако это действие с вызовом функции, набором переменных, циклом и условием уже больше похоже на настоящую программу. Вот пример ее запуска:

```
$ awk -f language.awk geodata
Албания
Греция
```

Одной командой покажем, что awk умеет выполнять арифметические действия – получим список стран с плотностью населения более 150 человек на квадратный километр:

```
$ awk ‘$3*1000/$2 > 150’ geodata
Люксембург 2586 465 Немецкий,французский Евро
Швейцария 41293 7252 Немецкий,французский,итальянский Франк
```

Обратите внимание, что в этой программе есть шаблон, но нет действия. Как видите, в этом случае используется действие по умолчанию – строка целиком выводится на экран.

Язык awk содержит массу возможностей, о которых здесь не рассказано, и многие пишут на этом языке программы гораздо длиннее, чем в четыре строчки! Впрочем, моя задача не в том, чтобы показать, какие длинные бывают программы, а в том, что и короткие программы делают немало полезного.

Пример с решением

Давайте объединим все, о чем мы говорили, в последний пример. Наша задача – подсчитать частоту появления слов в образце текста, а текст сегодня утром взят из Евангелия от Марка, Глава 6 (версия короля Якова; загрузить его можно из центра электронных текстов библиотеки Вирджинского университета – http://etext.virginia.edu/kjv.browse.html) [король заказал первый перевод Писания на англ. яз. – прим. ред.]. Теоретически мы можем ввести все решение с командной строки, но лучше оформить его в виде скрипта **wordfreq.sh**. Мы будем добавлять в этот файл по одной строке и контролировать результат работы скрипта на каждом этапе.

Каждая строка файла, который я скачал с сайта Вирджинского университета, соответствует одному стиху и начинается с его номера и двоеточия. Например, строка для стиха 42 выглядит так:

```
42: And they did all eat, and were filled.
```

[И все ели, и насытились.] Я сохранил этот текст в файле **mark.txt**.

Чтобы определить количество слов, воспользуемся ассоциативными массивами, но сперва надо слегка почистить входной текст. Для начала избавимся от этих номеров. Мы можем использовать команду замены редактора *sed*. Эта команда и будет первой строкой нашего скрипта **wordfreq.sh**:

```
#!/bin/bash
sed ‘s/^[0-9]*:\ //’ $1
```

Первая строка файла – это часть механизма скриптов в Linux. Она предписывает операционной системе использовать оболочку *bash* для интерпретации скрипта. Вторая строка – классический пример использования *sed*. Использование команды замены ясно из предыдущих примеров; «старый шаблон» использует регулярные выражения, соответствующие фрагменту «начало строки, далее возможны несколько цифр, затем двоеточие и пробел», а «новый шаблон» между вторым и третьим прямыми слэшами пуст. Таким образом, номера в начале строк удаляются. $1 в конце этой строки еще немного приоткрывает нам механизм скриптов в Linux – он будет заменен аргументом командной строки, который мы передадим скрипту. Это имя файла, который должен обработать *sed*.

Создав наш двустрочный скрипт, мы должны разрешить его выполнение:

```
chmod u+x wordfreq.sh
```

Теперь можно запустить скрипт, передав ему в качестве аргумента имя файла:

```
./wordfreq.sh mark.txt
```

Стих 42 из нашего примера теперь выглядит так:

```
And they did all eat, and were filled.
```

Затем я решил сделать подсчет слов нечувствительным к регистру. Проще всего для этого преобразовать все символы верхнего регистра в исходном тексте в нижний регистр, что легко делается командой tr. Теперь наш скрипт стал на три строки длиннее:

```
#!/bin/bash
sed ‘s/^[0-9]*:\ //’ $1 | \
tr ‘[A-Z]’ ‘[a-z]’
```

Мы добавили ко второй строке обозначение канала | и обратный слэш, означающий, что команда продолжится в третьей строке. На самом деле такие выражения не нужно разбивать на отдельные строки: я сделал это лишь затем, чтобы упростить чтение скрипта. Команда tr на третьей строке – тоже классика. Она означает «заменить каждый символ из набора A-Z соответствующим символом из a-z». Если запустить наш новый скрипт из трех строк, то строка из примера будет выглядеть так:

Перехват стандартного вывода

В большинстве наших примеров предполагается, что стандартный вывод связан с окном терминала (вариант по умолчанию). Однако перенаправить его в файл очень просто, использовав оператор оболочки >. Например, команда

```
$ sort foo.txt | tr ‘[A-Z]’ ‘[a-z]’ > sorted.txt
```

запускает ту же самую цепочку команд, что и раньше, но перенаправляет стандартный вывод stdout команды tr в файл **sorted.txt**. Заметьте: перенаправление выполняет сама оболочка, команда tr просто выводит данные в stdout и не знает, да и знать не хочет, куда направляются эти данные.

```
and they did all eat, and were filled.
```

На следующем шаге избавимся от этих нудных знаков препинания. Это тоже можно сделать с помощью команды tr (используя ключ -d). Итак, теперь наш скрипт будет выглядеть следующим образом:

```
#!/bin/bash
sed 's/^[0-9]*:\ //' $1 | \
tr '[A-Z]' '[a-z]' | \
tr -d '[.,;:]'
```

Последняя строка просто удаляет все символы из набора [.,;:]. После запуска этой версии скрипта наша строка выглядит так:

```
and they did all eat and were filled
```

Вот это уже можно отдать на съедение awk: именно здесь мы и определим частоту появления слов. Основная идея состоит в прокручивании каждого отдельного слова в документе, используя само слово как индекс в ассоциативном массиве и просто увеличивая на единицу соответствующий элемент этого массива. Обработав весь документ, мы сможем вывести индекс и значение каждого элемента массива – то есть само слово и сколько раз оно встречается в документе. Я решил записать программу в отдельный файл с именем **wordfreq.awk**; таким образом, сейчас у нас есть два скрипта, с которыми можно поработать – это скрипт оболочки **wordfreq.sh** и программа на языке awk **wordfreq.awk**.

Наш скрипт выглядит следующим образом:

```
sed 's/^[0-9]*:\ //' $1 | \
tr '[A-Z]' '[a-z]' | \
tr -d '[.,;:]' | \
awk -f wordfreq.awk
```

а программа на языке awk выглядит так:

```
{ for (i=1; i<=NF; i++)
   w[$i]++
}
END { for (word in w)
        print word, w[word]
}
```

awk-программа содержит два действия. Проще начать с первого, которое применяется ко всем строкам, так как не содержит шаблона. Это действие обрабатывает все поля во входной строке (т.е. все слова) и увеличивает на единицу соответствующий элемент ассоциативного массива w. Имена переменных i и w я выбрал сам, переменная NF – внутренняя переменная языка, которая содержит количество полей в текущей строке. Выражение w[$i]++, которое увеличивает на единицу соответствующий элемент ассоциативного массива w – главная часть этой программы. Все остальное лишь обеспечивает его работу.

Второе действие в этой программе срабатывает только один раз, после того, как все входные данные обработаны. Оно просто перебирает элементы массива w и выводит индекс элемента в массиве (само слово) и значение элемента (сколько раз это слово встретилось в тексте).

Вывод нашего скрипта в корне изменился и теперь выглядит так:

```
themselves 4
would 3
looked 1
taken 1
of 27
sit 1
privately 1
abroad) 1
name 1
and 134
```

…следующие строк этак 400 пропущены.

Наконец, мы можем вывести эти данные в более удобной форме. Для этого отсортируем их по числу вхождений (чтобы список начинался с чаще всего встречающихся слов) и применим head, чтобы оставить только 10 первых строк. Окончательная версия скрипта будет выглядеть так:

```
#!/bin/bash
sed 's/^[0-9]*:\ //' $1 | \
tr '[A-Z]' '[a-z]' | \
tr -d '[.,;:]' | \
awk -f wordfreq.awk | \
sort -nr -k2 | \
head
```

Обратите внимание на флаги команды сортировки. -n включает сортировку по численному значению, -r реверсирует результаты сортировки, а -k2 сортирует данные по значению второго поля. Теперь у нас есть желаемые данные о частоте появления слов:

```
and 134        of 27
the 64         him 26
he 38          unto 23
they 31        to 22
them 31        his 21
```

Сделайте это по-разному

Как и большинство вещей в жизни, наша задача решается разными способами. Вместо использования ассоциативных массивов awk можно разбить содержимое файла на отдельные слова (по одному слову на каждой строке) командой tr, затем упорядочить этот список по алфавиту (вхождения одного и того же слова будут расположены друг за другом), после чего командой uniq подсчитать, сколько раз появляется каждое слово, отсортировать список по убыванию и выделить первые десять строк. Такой скрипт будет выглядеть следующим образом:

```
#!/bin/bash
sed 's/^[0-9]*:\ //' $1 | \
tr '[A-Z]' '[a-z]' | \
tr -d '[.,;:]' | \
tr ' ' '\n' | \
sort | \
uniq -c | \
sort -nr | \
head
```

а результат его работы выглядит так:

```
134 and        27 of
64 the         26 him
38 he          23 unto
31 they        22 to
31 them        21 his
```

Он точно такой же, как и предыдущий, только поля расположены в обратном порядке. Чтобы понять, как работает эта программа, попробуйте создать ее по шагам (на каждом этапе вводите произвольные данные) и анализируйте результат ее работы. LXF

Скорая помощь

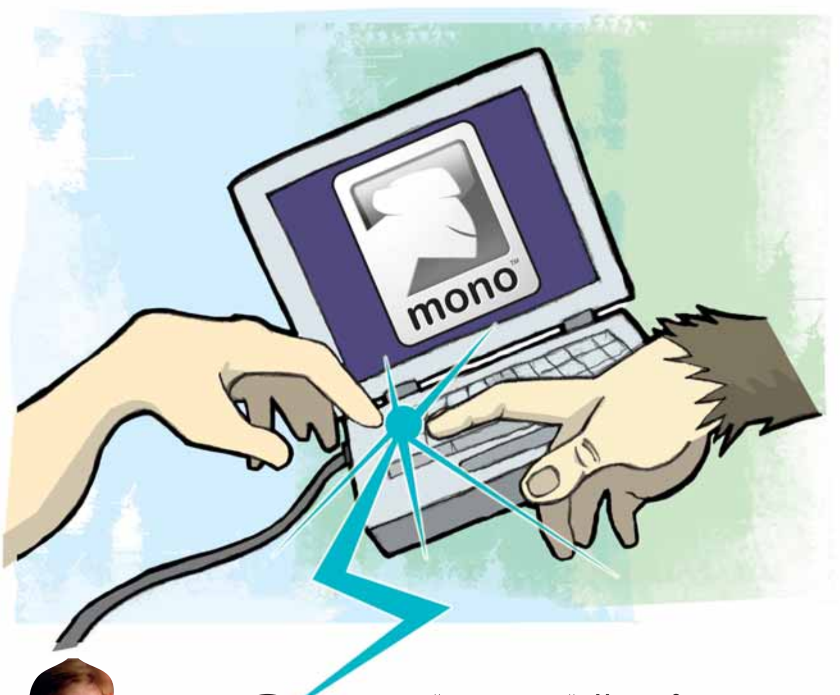
Чтобы подробно изучить регулярные выражения, найдите старый номер LXF80 и откройте стр. 74. Или познакомьтесь с формальным описанием их синтаксиса на странице man regex (понять его практически невозможно).

Регулярные выражения

Команда	Что она делает
head /etc/passwd	Показывает первые десять строк файла **/etc/passwd**.
grep '/bin/bash$' \ /etc/passwd	Показывает строки из **/etc/passwd** о пользователях, использующих *bash* для входа в систему.
sort /etc/services	Сортирует сервисы из файла **/etc/services** в алфавитном порядке.
wc /etc/* 2> /dev/null	Считает строки, слова и символы во всех файлах каталога **/etc**; сообщения об ошибках игнорируются.

Mono: Шифруем

Если вы не параноик, это не значит, что за вами не следят. **Пол Хадсон** создает шифровальщик файлов правительственного уровня, пока он еще может...





Наш эксперт

Пол Хадсон полагает, что Mono – лучшая вещь со времен мультфильма Pinky and the Brain, и сейчас поддерживает два проекта на основе Mono на SourceForge.

Со времен тайных посланий Марии Стюарт, королевы Шотландской, про свержение королевы Елизаветы, и до системы «Энигма» в нацистской Германии, давно уже ясно, что все, что шифруется, можно и расшифровать – были бы инструменты. Марию сгубил союз сэра Фрэнсиса Уолсингема [Francis Walsingham] и Томаса Феллипса [Thomas Phelippes]: им было поручено разгадывать ее переписку, защищая королеву Елизавету от заговорщиков. Для машины Энигма роковой стала череда роковых случайностей, остроумных догадок и приемов математики. Но итог в обоих случаях один: сообщения, посылаемые в расчете на секретность, были взломаны и превратились в обычный текст.

Я рассказываю все это как предисловие к проекту этого месяца, потому что мы собираемся создать программу шифрования и дешифровки файлов. Для проекта мы применим супер-мощный Advanced Encryption Standard [Продвинутый стандарт шифрования], который оценен Национальным агентством безопасности США как пригодный для документов высокой секретности, но не обольщайтесь: уж если крупная правительственная организация захочет взломать ваши данные, она их взломает. К счастью, я уверен, что скрывать вам нечего, а значит, шифрование, изученное на этом уроке, вполне подойдет, чтобы не дать случайным зевакам пялиться в ваши данные, которые им незачем видеть!

Наш проект этого месяца будет шифровать и дешифровать файлы, основываясь на заданном вами секретном ключе шифрования – это 256-битный ключ AES, способный защитить от хакеров практически любой масти. Как вы, надеюсь, знаете, «бит» – это одна восьмая байта, так что 256-битный ключ эквивалентен 32-м байтам данных. Так уж вышло, что один байт – это ровно столько, сколько требуется для хранения одного ASCII-символа; значит, наш ключ – это пароль из 32-х символов.

Если вы не Джеймс Бонд, то, скорее всего, у вас нет 32-символьного пароля – в лучшем случае 12 символов (если вы особо заботитесь о безопасности), но наиболее вероятно – около 8. Как же создать такой 32-символьный AES-ключ, чтобы вы его не забыли или, упаси Боже, не записали его? Есть два возможных ответа:

1. Дополнить ваш пароль до 32 символов. Например, если это frosties, преобразуем его в frosties000000000000000000000000. Надеюсь, вы понимаете, почему это плохое решение!
2. Создать хэш-сумму ключа. Это преобразует frosties в, на первый взгляд, случайный поток символов и цифр. Замечательное свойство хэш-ключа в том, что, подавая frosties на вход хэш-алгоритма, мы всегда получаем одно и то же.

Второй вариант, очевидно, лучше, так что наша первая задача – прочесть секретный ключ пользователя и преобразовать его в хэш. Самый популярный хэш-алгоритм – Secure Hash Algorithm [Защищенный хэш-алгоритм], но он не столь защищен, как многие о нем думают; есть более новые алгоритмы SHA-256 и SHA-512, выполняющие тут же задачу с большей надежностью. Конечно, проблема с любым новым алгоритмом в том, что его защищенность не доказана, так что пока мы удовольствуемся старым добрым SHA1.

Создаем хэш

Среда .NET предоставляет на выбор несколько хэш-алгоритмов; используются они примерно одинаково. Для SHA1 необходимо создать объект SHA1Managed, передать ему строку для хэширования как поток байт, затем прочесть поток байт хэшированного значения и преобразовать его в строку. Возможность генерации хэша для любых входных данных исключительно полезна; я создам небольшой метод Sha1(), которому можно передать строку текста и получить хэш-строку, приблизительно такой:

```
static string Sha1(string input) {
  SHA1Managed hashgen = new SHA1Managed();
  byte[] hash = hashgen.ComputeHash(Encoding.UTF8.GetBytes(input));
  string result = "";
  foreach (byte b in hash) result += b.ToString("x2");
  return result;
}
```

!!! ПРЕДОСТЕРЕЖЕНИЕ !!!

На этом уроке мы занимаемся шифрованием файлов с использованием указываемого вами секретного ключа. Если вы передадите этот ключ другим людям, они смогут расшифровать ваши файлы. Если вы его забудете, то не сможете извлечь информацию из файлов, зашифрованных им. Вы предупреждены!

Месяц назад Мы сделали клиент мгновенных сообщений для локального или web-чата.

LINUX FORMAT На DVD

» Mono 1.2.5
» MonoDevelop 0.14 и 0.15
» Код учебника

ваши файлы

Здесь два действительно сложных места: это преобразование ввода в последовательность байт при помощи Encoding.UTF8.GetBytes() и преобразование хэшированных байтов обратно в шестнадцатеричную строку через b.ToString("x2"). Но они сложны лишь потому, что вы, вероятно, видите их впервые – не так все страшно, когда они упакованы в опрятный метод, трогать который вам больше не придется!

Чтобы попользоваться прелестями этой криптографии, вам понадобится добавить следующие строки в заголовок любого кода, который будет использовать упомянутый метод Sha1():

```
using System.Security;
using System.Security.Cryptography;
using System.Text;
```

Вперед!

Итак, теперь, когда вы поняли, как работает хэш, я хочу, чтобы вы запустили MonoDevelop и создали новый проект с именем *Tanuki* (если вы распознаете намек, я немало удивлюсь). Итоговый метод Main() этого проекта должен читать имя файла и параметр + или – (для шифрования или дешифрования файла), а также строку текста из командной строки, которая передастся функции Sha1() для создания ключа шифрования. Но сначала убедимся, что система Sha1() корректно работает, заставив Main() прочесть секретный ключ, хэшировать его и затем вывести результат на экран.

```
static void Main(string[] args) {
 // Мы хотим прочесть секретный ключ, так что читаем некий текст и помещаем его в переменную "key"
  Console.Write("Пожалуйста, введите ключ: ");
  string key = Console.ReadLine();
  // Если ключ не указан, использовать ключ по умолчанию
  if (string.IsNullOrEmpty(key)) {
    Console.WriteLine("Ключ не указан – считаем, что это 'frosties'");
    key = "frosties";
  }
  // Создание хэша ключа
  key = Sha1(key);
  Console.WriteLine(key);
}
```

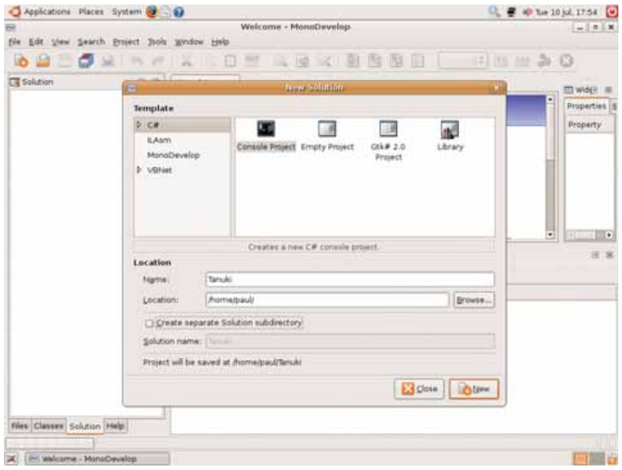
Скиньте это в ваш новый проект, вместе с методом Sha1(), описанным ранее, и вы готовы к бою: попытайтесь запустить его, чтобы убедиться, что ваш генератор SHA1 работает нормально.

Секретное послание

Благодаря ReadLine() и Sha1() теперь у нас есть 40-битный ключ безопасности, 32 бита которого будут использоваться для шифрования наших файлов. Здесь начинается реальная сложность: зашифровать и расшифровать файлы с использованием .NET не так легко, и это странно, для столь широко используемой операции! Фактически, шифрование и расшифровка, вместе с использованием сети (как показал предыдущий номер), это уродливые наросты, которые, похоже, необходимы любой среде программирования, чтобы считаться полноценной.

Труднейшая часть шифрования – работа с данными, пока они блуждают между входным и выходным файлами. Вообще-то создать AES-шифровальщик предельно просто, раз у нас уже есть готовый к применению ключ. Фактически это сводится к следующему:

```
RijndaelManaged aes = new RijndaelManaged();
aes.Key = Encoding.UTF8.GetBytes(key.Substring(0, 32));
aes.IV = Encoding.UTF8.GetBytes(key.Substring(5, 16));
```



» Это лишь еще один проект для командной строки, так что убедитесь, что вы выбрали верные опции в *MonoDevelop*. Как обычно, не нужно позволять ему создавать отдельный подкаталог.

Rijndael – это прежнее название алгоритма AES, до того, как он был принят в качестве стандарта. Заметьте, что мы используем Substring() для чтения первых 32-х байт ключа SHA1, поскольку это все, что нам нужно. IV – сокращение от Initialisation Vector [Вектор инициализации], и может рассматриваться как беглое предварительное шифрование перед основным шифрованием. Прогон входных данных через IV известен как «отбеливание» [от «белый шум», – прим. ред.] – перед зашифровкой ваш текст преобразуется в нечто более безликое, и если даже кто-то доберется до вашего ключа перебором миллионов комбинаций, он не будет знать, достиг ли он верного результата, потому что IV его исказил. Для AES, IV должен состоять из 16 символов, так что мы читаем из ключа SHA1 с 5-го символа по 21-й, просто для краткости.

Итак, этот кусок весь довольно простой. Более трудная часть – протащить входной файл через шифровальщик в выходной файл, поскольку это выполняется при помощи файловых потоков. Делается это приблизительно так:

- Открыть файловый поток с правами записи; он будет использоваться для сохранения результата
- Создать движок Rijndael
- Создать базовый шифровальщик, указав ему наш движок Rijndael и выходной файловый поток
- Прочитать входной файл в массив байтов
- Передать массив байтов шифровальщику, для его шифрования и записи в результат
- Закрыть открытые файлы

Затем, в завершение, засунем весь этот блок кода в приятный метод EncryptFile(), чтобы больше никогда его не видеть!

Превращая все это в C#, получаем следующий большой метод, с комментариями для удобства чтения:

```
static void EncryptFile(string file_in, string file_out, string key) {
 // Я заключил весь метод в блок try/catch для перехвата ошибок
  try {
FileStream output = new FileStream(file_out, FileMode.Create,FileAccess.Write);

       // Это тот же код, что и ранее
     RijndaelManaged aes = new RijndaelManaged();
     aes.Key = Encoding.UTF8.GetBytes(key.Substring(0, 32));
     aes.IV = Encoding.UTF8.GetBytes(key.Substring(5, 16));
```

»

Скорая помощь

Если вы хотите самостоятельно опробовать SHA512, просто замените класс SHA1Managed на SHA512Managed. Но помните, что это приведет к генерации ключей, отличающихся от сгенерированных SHA1!

```
  // Этот кусок делает всю трудную работу
  ICryptoTransform transform = aes.CreateEncryptor();
  CryptoStream cryptostream = new CryptoStream(output,transform,
CryptoStreamMode.Write);
  // Вытягиваем входной файл...
  byte[] inputbytes = File.ReadAllBytes(file_in);
  // ...и проводим его через CryptoStream в выходной файл
  cryptostream.Write(inputbytes, 0, inputbytes.Length);
  // а теперь очищаем
  cryptostream.Close();
  output.Close();
 } catch (Exception e) {
  // При возникновении проблем (не найден файл и т. п.), печатаем
ошибку
  Console.WriteLine(e.Message);
 }
}
```

Необходимость использовать массивы байтов смахивает на откат к C, но это единственный способ корректно работать с двоичными данными. Если вы попытаетесь сохранить шифрованные данные в виде строк, то почти наверняка повредите ваши данные – будьте осторожны!

Подставляем другую щеку

Шифрование файлов без возможности их дешифрации имеет смысл, только если вы для хранения бумаг регулярно применяете шредер. Но цель *Tanuki* – выполнять шифрование и дешифрацию файлов, используя один и тот же ключ – означает, что нам необходим метод Decrypt(), в пару к вышеприведенному Encrypt().

Дешифрация файлов немного отличается от шифрования, потому что нам необходимо читать дешифрованные данные и записывать простой текст. Это может быть сделано путем прямого подключения выхода шифрованного потока на вход нашей операции записи, почти как “>” в командной строке. Лучший способ объяснить это – предоставить вам код с комментариями, где происходит изменение; вот он:

```
static void DecryptFile(string file_in, string file_out, string key) {
 try {
  // Готовимся к чтению со входа и к записи на выход
  FileStream input = new FileStream(file_in, FileMode.Open,FileAccess.
Read);
   StreamWriter output = new StreamWriter(file_out);
   RijndaelManaged aes = new RijndaelManaged();
   aes.Key = Encoding.UTF8.GetBytes(key.Substring(0, 32));
   aes.IV = Encoding.UTF8.GetBytes(key.Substring(5, 16));
   ICryptoTransform transform = aes.CreateDecryptor();
   CryptoStream cryptostream = new CryptoStream(input, transform,
CryptoStreamMode.Read);
   // Записываем все, что прислал дешифровщик CryptoStream
   output.Write(new StreamReader(cryptostream).ReadToEnd());
```

Скорая помощь

Мы используем Encoding.UTF8.GetBytes() для преобразования нашей строки-ключа в последовательность байтов, но технически это не обязательно, потому что Sha1() возвращает лишь ASCII-символы, и можно обойтись Encoding.ASCII.GetBytes(). Однако в идеале мы не хотим, чтобы Encrypt() или Decrypt() вникали, в каком формате они получают данные, поэтому работа с UTF8 (известной также как Unicode) – более умное решение.

Забираемся глубже!

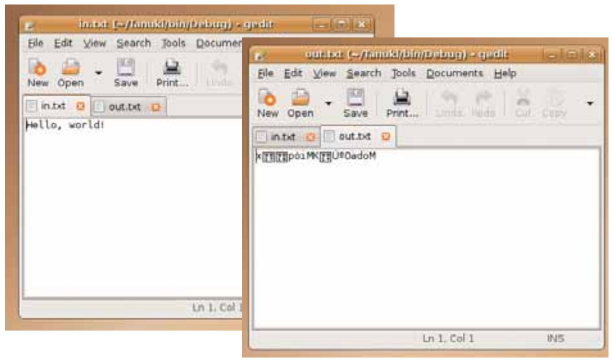
Если вы закончили *Tanuki* и ищете способы набраться опыта, попробуйте такие идеи:

» Легко: Заставьте *Tanuki* записывать секретный журнал с информацией о том, когда каждый файл был зашифрован или дешифрован.

» Средне: Позвольте людям выбирать другие алгоритмы шифрования, а не только AES.

» Трудно: Позвольте пользователям указывать каталоги для шифрования. Пусть выходной файл автоматически приобретает вид **файл.расширение.encrypt**, т. е. **/foo/bar/baz.txt** превратится в **/foo/bar/baz.txt.encrypt**

» Экстрим: Кэшируйте секретные ключи пользователей на 10 минут, чтобы им не приходилось постоянно вводить их при шифровании различных файлов. Подсказка: наиболее безопасный метод требует другой программы, запущенной как демон, а также связи через сокеты.



› Файл на входе... › ...и на выходе, после однократного прогона через Tanuki.

```
  // Убеждаемся, что запись закончена, затем закрываем дескриптор
файла
   output.Flush();
   output.Close();
 } catch (Exception e) {
   Console.WriteLine(e.Message);
 }
}
```

Как и при шифровании, я умышленно поместил расшифровку в отдельный метод – вы можете вставить этот код в любой проект, и он просто будет работать.

Соединяем все вместе

Теперь мы умеем шифровать файлы, дешифровывать файлы и генерировать хэши строк для использования в качестве секретных ключей; но *Tanuki* пока что лишь набор методов, гуляющих сами по себе. В некотором смысле это неплохо: работа с такими самодостаточными кодами означает, что у вас есть набор строительных блоков, которые можно поместить куда угодно. Более того, если когда-нибудь вы решите изменить ваш алгоритм с AES на какой-то другой, останется лишь изменить несколько строк в Encrypt() и Decrypt() – остальной код менять не нужно, потому что он и знать не знает, как работают Encrypt() и Decrypt(). Этот метод, известный как «инкапсуляция» или «скрытие данных», является фундаментальной концепцией программирования.

Итак, на данном этапе нам осталось обновить метод Main(), чтобы он читал параметры и выполнял соответствующие действия. Параметры, которые должен воспринимать *Tanuki* – это указание шифровать или дешифровать файл (при помощи + или –), а также имена входного и выходного файлов. В общем, метод должен:

1. Проверить, налицо ли все три параметра (и выкручиваться, если их окажется больше/меньше).
2. Прочесть секретный ключ пользователя из командной строки (и использовать frosties, если ничего не было введено).
3. Создать хэш ключа.
4. Если первый параметр +, шифровать файл.
5. Если первый параметр –, дешифровать файл.

Преобразуя это в комментированный код, получим:

```
static void Main(string[] args) {
 // Нам нужно ровно три аргумента!
 if (args.Length != 3) {
   Console.WriteLine(“Укажите либо +, либо - как первый параметр,
потом введите  входной и выходной файлы как второй и третий
параметры.”);
   return; // Выкручиваемся
 }
 // Это уже было
 Console.Write(“Пожалуйста, введите ключ: “);
 string key = Console.ReadLine();
 if (string.IsNullOrEmpty(key)) {
```

Пароли как параметры

У вас может быть искушение позволить людям запускать *Tanuki* так:

```
mono tanuki.exe + in.txt out.txt my_secret_key
```

На вид более удобно, и это вправду удобно. А заодно и ужасно небезопасно, поскольку любой может просто влезть в вашу историю *Bash* и узнать пароль. Единственный способ обойти это – очистить историю (выполнив history -c), но проще запросить пароль после запуска программы.

```
    Console.WriteLine("Ключ не указан – считаем, что это 'frosties'");
    key = "frosties";
  }
  key = Sha1(key);
  // Выбираем, что делать, согласно первому параметру
  if (args[0] == "+") {
    EncryptFile(args[1], args[2], key);
  } else {
    DecryptFile(args[1], args[2], key);
  }
}
```

В использовании программой параметров имеется один существенный недостаток – рекомендую вам попробовать самим его обнаружить, пока я не назвал его. Покамест попытайтесь запустить вашу программу так:

- Создайте файл с именем **foo.txt** в вашем каталоге **bin/Debug**.
- Поместите в него что-нибудь. Но только не много.
- Теперь выполните: mono tanuki.exe + foo.txt foo-enc.txt
- Затем: mono tanuki.exe - foo-enc.txt foo-copy.txt

Если все работает, то **foo.txt** и **foo-copy.txt** должны совпасть.

Нашли проблему? Нет? Хорошо, даю вам еще минуту, и пользуюсь случаем напомнить вам, что для максимальной безопасности понадобится вектор инициализации получше. У использованного мною IV есть проблема: он поставляется из того же источника, что и ключ (обработанная SHA1 версия текста, предоставленного пользователем). Все потому, что я использую одну и ту же часть хэша для создания ключа и IV, вместо того, чтобы использовать для IV несколько символов не из ключа. Помните, однако, что IV не должен храниться в секрете, потому что его основная работа – малость перемешать ваш простой текст перед тем, как он отправится на шифрование.

Работа над ошибками

Я думаю, что у вас было достаточно времени осознать проблему с параметрами в *Tanuki*, а потому даю ответ: если args[0] равен «+», мы шифруем файлы. В противном случае, мы их дешифруем. А вдруг кто-то случайно перепутает порядок следования параметров, т.е.:

```
mono tanuki.exe in.txt out.txt +
```

В этом примере *Tanuki* попытается дешифровать файл, а затем вылетит, потому что мы указали в качестве имени выходного файла +. Простейший способ исправить этот казус – ввести проверку, например, так:

```
if (args[0] == "+") {
  EncryptFile(args[1], args[2], key);
} else if (args[0] == "-") {
  DecryptFile(args[1], args[2], key);
} else {
  Console.WriteLine("Укажите либо +, либо - как первый параметр");
}
```

Для начала неплохо, но это только часть проблемы. Что произойдет, если входной файл не существует? Что произойдет, если выходной файл существует? Проверка первого может быть выполнена путем простого добавления следующего кода сразу после проверки args.Length:

```
if (!File.Exists(args[1])) {
  Console.WriteLine("Входной файл не существует!");
  return;
}
```

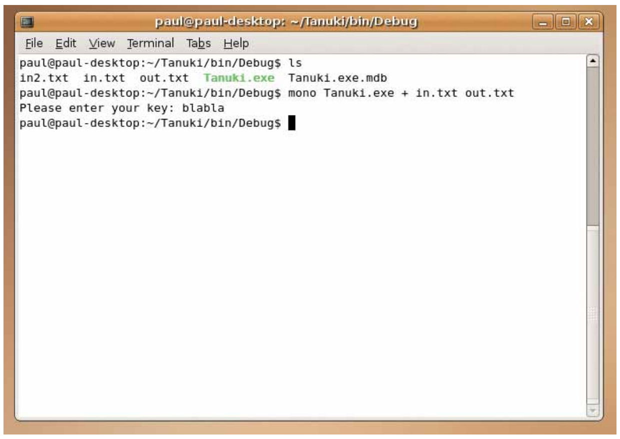
Что касается второй проблемы, ее решение зависит от того, как вы хотите с ним работать. Я специально написал *Tanuki* так, чтобы она работала с входным и выходным файлами, и никакая ошибка, случайно допущенная вами при вводе кода, не уничтожила ваши драгоценные файлы! Но коль скоро ваш код заработал, почему бы вообще не отбросить параметр выходного файла и не выполнять шифрование прямо на месте. Как альтернатива, вы можете позволить людям запускать Tanuki с параметрами + и – (шифрование и дешифрация) или ++ и — (шифрование и дешифрация, и не выкручиваться, если выходной файл уже существует).

Итак, мы подошли к концу проекта этого месяца, и я надеюсь, вы видите, что теперь мы получили действительно серьезный код. Если даже вам не на 100% ясно, как работают Encrypt(), Decrypt() и Sha(), фишка в том, что они существуют как закрытые методы, о которых вам нечего беспокоиться – они работают, вот и все! В частности, я счел метод Sha1() полезным в моем собственном коде, поскольку создание хэш-значений (всего, от паролей до DVD-образов в 8 ГБ) является прекрасным способом проверки корректности полученных данных. Но в любой ситуации, когда вы не желаете, чтобы мир копался в ваших данных, шифрование – лучший выбор.

Наслаждайтесь – шифрование ваших файлов поможет вам быть на шаг впереди силовых ведомств! LXF

Скорая помощь

Скомпилированную программу Mono очень просто преобразовать обратно в более или менее понятный код, в основном потому, что большинство .NET программ содержит много легко распознаваемых встроенных функций. Так что не ожидайте выигрыша от любых усилий вроде «неясно – значит, безопасно»: Tanuki защищена, потому что защищен AES и потому что секретный ключ хранится в секрете, а не потому, что вы припрятали исходный код только для себя.



› *Tanuki* в действии. Есть несколько вещей, которые вы можете захотеть попробовать или исправить: каждый может видеть секретный ключ при его вводе (он не скрыт за звездочками), и нет сообщений подтверждения.

» **Через месяц** Встреча Mono и Posix: как использовать любимые функции Linux.

Gentoo: Не жди

Необходимо самое свежее ПО, еще не попавшее в portage? **Нейл Ботвик** покажет пользователям Gentoo, как отследить ebuild или написать свой собственный.





Наш эксперт

Нейл Ботвик считает, что Gentoo – лучший дистрибутив после его предыдущего любимца, и никогда не упускает возможности исправить то, что не было сломано.

Кроме рассылки своевременных ответов на вопросы озадаченных пользователей через форумы www.linuxformat.co.uk, наибольшее удовлетворение от работы в Linux Format нашей команде приносят ответы на избранные запросы наших читателей, присылаемые на наши адреса (*см. стр. 107*). Мы хотим лишь одного: чтобы у нас было побольше времени на помощь!

Уважаемый Linux Format,

В вашем обзоре Gentoo сказано, что в нем почти 12 000 пакетов, и они всегда свежие, так почему же я не могу установить новую версию *Snafu 2.0*?

Брюзга из киберпространства

Уважаемый/ая г-н/г-жа Брюзга,

Действительно, пакетов для Gentoo очень много: сейчас есть более 24 000 ebuild'ов для более 12 000 пакетов; но нет такого репозитория, где содержится абсолютно все. Однако Вы можете заполучить Snafu хоть сию же минуту, причем не одним способом. Так что давайте рассмотрим некоторые способы его найти, или даже, если ни один не поможет, напишем свой собственный ebuild, что не так страшно, как Вам может показаться. Заодно мы рассмотрим несколько полезных программ, которые делают работу с *portage* еще легче.

Я случайно подметил, что Вы пожаловались лишь на сравнительно малоизвестную программу *Snafu*; более известные *foo* и *bar* всегда идут в ногу со временем, поскольку они необходимы для корректной работы большинства man-страниц. Читайте... **НБ**

Шагать в ногу со временем

Если последнюю синхронизацию *portage* вы проводили раньше, чем вчера, то запустите emerge --sync, чтобы убедиться в наличии всех свежих обновлений, содержащих то, что вам необходимо. Если *Snafu 2.0* все еще нет и он был выпущен менее чем месяц назад, весьма вероятно, что он уже есть в *portage*, но помечен для тестирования. Большинство ebuild'ов находятся в дереве около 30 дней, прежде чем их пометят как стабильные; очевидное исключение – исправления безопасности, которые поставляются всем при первой же возможности. Управляет этим параметр KEYWORDS,

```
KEYWORDS="x86 ~amd64"
```

означающий, что ebuild рассматривается как стабильный для x86, но все еще тестируется для amd64. И вам не нужно закапываться в ebuild'ы, чтобы это обнаружить! Первая полезная программа уже здесь, она называется *eix*, вот ее и emerge'ните. Это программа поиска в *portage*, она в несколько раз быстрее, чем использование emerge --search, и к тому же предоставляет больше информации. Быстротой она обязана использованию индексов, так что после ее установки запустите update-eix. Базу данных необходимо обновлять каждый раз при запуске emerge --sync; программа *eix-sync* позаботится об обеих задачах и сообщит вам о новых и обновленных пакетах. Теперь выполните eix некая_программа, и увидите, что есть в *portage*. Вывод *eix* четко различает стабильные, тестовые и маскированные (об этих далее) пакеты. Версия пакета, устанавливаемого portage, управляется параметром ACCEPT_KEYWORDS в вашем **/etc/make.conf**, но его можно назначить самим. При желании установить тестовый пакет, выполните

```
ACCEPT_KEYWORDS="~amd64" emerge snafu
```

но это должно использоваться только с --pretend, для тестирования. Имеются по меньшей мере две веские причины не использовать ACCEPT_KEYWORDS в командной строке. Во-первых, это будет применено ко всем устанавливаемым пакетам: не только к *Snafu*, но и к его зависимостям. Во-вторых, установка является временной, и при следующем emerge world *portage* попытается вернуть все назад. Решение – в **etc/portage/package.keywords**. Файлы в **/etc/portage** отменяют для отдельных пакетов настройки в **make.conf** и профиле *portage*. Итак, поместите следующую строку

```
net-misc/snafu ~amd64
```

в **/etc/portage/package.keywords** (если файл не существует, создайте его), и будете всегда использовать тестовую версию *Snafu*. Можно выполнить более тонкую настройку, применив ее к конкретной версии; можно предпочесть тестовую версию *Snafu 2.0* и остановиться на ней, когда она станет стабильной, а не гнаться постоянно за новыми версиями; более детально см. врезку под заголовком *«Атомная сила»* на следующей странице.



Месяц назад Применяли APT.

ebuild'ов!



› http://bugs.gentoo.org: запрос обновленных версий, отчет об ошибках и ваш успех в обновлении или размещении собственных ebuild.

Возможно также, что пакет просто не был протестирован на вашей архитектуре, и ebuild не содержит ключевых слов для нее. Обычно это случается со второстепенными архитектурами, но порой имеет место и для amd64. Часто вы можете заставить его установиться, изменив ACCEPT_KEYWORDS на значение ~x86 или x86 для этого пакета. Если таким образом все устанавливается и корректно запускается, отправьте отчет на http://bugs.gentoo.org, чтобы осуществляющие поддержку знали это. Я так поступал много раз, когда устанавливал на свой iBook программы, не тестированные для PPC.

Другой вариант – пакет замаскирован; тогда присвоение ему ключевых слов ни при чем. В этом случае пакет (обычно) приведен в списке **/usr/portage/profiles/package.mask**, вместе с причинами маскирования. Причина может быть любой: от «замаскирован для тестирования» до «подпалит вашего кота, если установите». Если вы готовы к подобным последствиям, добавьте атом пакета в **/etc/portage/package.unmask**, чтобы установить его.

Прежде чем оставить тему тестовых и стабильных ebuild, важно уяснить, что эти термины означают. Они соответствуют самому ebuild. Стабильный ebuild тестировался около месяца и потому стабилен, и в смысле доказательства работоспособности и в смысле других определений стабильности, и неизменяем. Это важное свойство стабильных пакетов: вам не надо обновлять вашу машину бесконечно, что важно на производстве. Тестовый, с другой стороны, не означает нестабильный, в смысле склонный к сбоям: это означает более быстрое изменение. Очень важно, что эти термины применяются к ebuild, но не к программам, которые они устанавливают. Хотя ebuild *Snafu 2.0* еще не помечен как стабильный, это не означает, что сама Snafu считается нестабильной, склонной к повреждению вашего компьютера или вызывающей выпадение волос.

Забираемся глубже

Хотя дерево *portage* – это главный репозиторий ebuild'ов, он отнюдь не единственный. *Portage* позволяет добавлять оверлеи, содержащие больше ebuild'ов. Если вы добавите строку

```
PORTDIR_OVERLAY=/usr/local/portage
```

в **/etc/make.conf** и создадите этот каталог, вы сможете копировать ebuild'ы в него. Где их найти? Первым делом обратитесь на сайт Gentoo, где регистрируются ошибки [bug tracker] – http://bugs.gentoo.org. Поищите по имени программы с префиксом ALL, чтобы убедиться, что просмотрены будут все ошибки, а не только открытые – например, ALL snafu. Вы можете обнаружить, что кто-то уже написал ebuild или изменил существующий, если он обновляется. Если нет, можете зарегистрировать ошибку с запросом на ebuild. Если вы ищете обновление программы, находящейся в *portage*, то дайте разработчикам время отреагировать на ваш запрос – не размещайте запросов на обновления через час после выхода программы. После *Bugzilla*, посмотрите на домашней странице программы – некоторые авторы выпускают свои собственные ebuild'ы; а не то поищите на форумах Gentoo.

Найдя ebuild, скопируйте его в соответствующее место вашего оверлея; при именовании должны использоваться стандартные категории, определенные для portage, так что ваш ebuild *Snafu 2.0* следует разместить в **/usr/local/net-misc/snafu/snafu-2.0.ebuild**. Если вы не смогли назвать его правильно, то portage, вероятно, проигнорирует его. Обычные предостережения касаются программ, поступающих из неофициальных источников: если они что-то сломают, вам останутся осколки и ничего более.

> «Древо Portage – это главный репозиторий ebuild'ов, но не единственный.»

Скорая помощь

Чтобы облегчить себе жизнь, не используйте номер версии внутри ebuild. Переменные вроде $P – все они описаны в man-страницах ebuild – делают ebuild переносимым между версиями одного и того же пакета.

Обновление существующего ebuild'a

Если вы собираетесь обновить существующий пакет, часто вы можете установить его, сделав копию существующего ebuild'a. Поскольку ebuild'ы используют другие файлы внутри каталога пакета, наилегчайший путь сделать это – скопировать весь каталог пакета с /usr/portage в ваш оверлей, затем скопировать ebuild, например:

```
cp -a /usr/portage/net-misc/snafu /usr/local/portage/net-misc/
cd /usr/local/portage/net-misc/
cp snafu-1.5-r1.ebuild snafu-2.0.ebuild
```

Выберите для копирования последний из существующих ebuild'ов. Часто ebuild'ы не требуют какой-либо правки, благодаря способу, »

Атомная сила

В документации ebuild вам встретятся упоминания об атомах; но что это такое? Атом – это спецификация какого-нибудь пакета, типа snafu или net-misc/snafu. Первый вариант можно использовать в командной строке, но не в других местах, потому что пакеты с тем же именем могут оказаться в других категориях: пример – emerge --pretend fuse. Значит, при работе с ebuild'ами нужно кроме имени указать категорию. Но иногда необходимо быть еще более точным. Все нижеследующее – это корректные атомы:

```
=net-misc/snafu-2.0
>net-misc/snafu-2.0
<net-misc/snafu-2.0
~net-misc/snafu-2.0
```

Первый соответствует только версии 2.0, второй соответствует только пакетам с версией выше 2.0 (чтобы включить 2.0, используйте >=), < и <= соответствуют меньшим номерам.

Последний пример менее привычен и означает 2.0 и любое исправление 2.0, такое как net-misc/snafu-2.0-r1, но не 2.0.1 или 1.9. Поскольку некоторые используемые здесь символы имеют особое значение при работе в оболочке, не худо бы выработать привычку заключать подобные атомы в кавычки.

» которым portage получает информацию о версии из имени ebuild. Скопированный ebuild будет знать, что нужно загрузить snafu-2.0.tar.bz2 со страницы проекта на SourceForge. Если вы сделаете это и он заработает, пожалуйста, сообщите об успехе разработчикам на http://bugs.gentoo.org – пусть знают, что он работает. Прежде чем работать с пакетом, добавленным вами к вашему оверлею, следует создать для него файл манифеста. Он содержит хэши MD5 и SHA1 всех файлов в каталоге пакета (а файлы в официальном дереве имеют также GPG-подпись). Это сделано для предотвращения вторжений malware под видом ebuild и означает, что protage откажется устанавливать пакет, если детали его манифеста не совпадают. Для ваших оверлейных ebuild'ов вам необходимо создать манифест самостоятельно, при помощи

```
ebuild /usr/local/portage/net-misc/snafu-2.0.ebuild manifest
```

Скорая помощь

Спустя какое-то время, файлы в **/etc/portage** захламляются избыточными или устаревшими элементами. Выполните eix-test-obsolete, чтобы увидеть «засохшие» ветки, которые можно отстричь.

Сторонние оверлеи

Оверлей portage вы можете создать сами, а можете во множестве импортировать их из других мест. Когда-то для этого приходилось использовать cvs или subversion, а затем редактировать /etc/make.conf, но сейчас у нас есть layman – один из полезных инструментов, о которых я говорил ранее: пришла пора его установить. Готово? Теперь выполните

```
layman -L
```

для загрузки и отображения последнего списка оверлеев. Список, правда, не слишком информативен: как узнать, какой оверлей содержит требуемый пакет? Вспомним eix. Он умеет загружать индексы также и из layman: выполните

```
update-eix-remote
```

и при следующем поиске с использованием eix выведется также и список пакетов, найденных в оверлеях. Добавьте этот оверлей в вашу систему так:

> «Слоты – это способ установки двух, обычно несовместимых, версий одного и того же пакета.»

```
layman -a overlayname
```

Когда вы делаете это в первый раз, вам следует добавить следующую строку в /etc/make.conf

```
source /usr/portage/local/layman/make.conf
```

Она известит portage обо всех добавленных layman оверлеях, так что вы сможете устанавливать из них через emerge. Один из первых рассматриваемых оверлеев – Sunrise (http://overlays.gentoo.org/proj/sunrise), содержащий избранные ebuild'ы с http://bugs.gentoo.org, он ближе всех к официальному оверлею.

Ваш первый ebuild

Вы пытались переименовать старый ebuild (если это обновление), искали в http://bugs.gentoo.org, http://forums.gentoo.org, в Google и под кроватью, но ebuild для вашего пакета так и не нашелся. Есть еще одна возможность: написать ebuild самостоятельно. Спешу вам сообщить, пока вы в ужасе не отбросили журнал, что написание ebuild для Gentoo намного проще, чем создание RPM- или Deb-пакетов, потому что большую часть работы выполняет portage. Если пакеты для сборки и установки используют стандартный автоматический процесс *./configure && make && make install*, то процедура практически тривиальна.

Вы найдете шаблоны ebuild в **/usr/portage/skel.ebuild**, они почти целиком состоят из пояснительных комментариев. Ebuild – это скрипт *bash*, управляющий загрузкой, распаковкой, конфигурированием, компиляцией и установкой программ. *Portage* использует множество функций для выполнения этих задач, главные из которых – src_fetch(), src_compile() и src_install(). Вы можете определить их в вашем ebuild, но если этого не сделать, то определения по умолчанию работают в стандартном случае *./configure && make && make install*. Получается, что простейший ebuild – просто набор присвоений значений переменным, вроде

```

# Copyright 1999-2007 Gentoo Foundation
# Distributed under the terms of the GNU
General Public License v2

# $Header: $
DESCRIPTION="Snafu is a great program"
HOMEPAGE="http://www.snafu.con/"
SRC_URI="http://www.snafu.con/downloads/${P}.tar.bz2"
LICENSE="GPL-2"
SLOT="0"
IUSE=""
KEYWORDS="~x86"

```

Три первые строки необходимы, если вы отправляете свой ebuild на http://bugs.gentoo.org. Остальные переменные обязательны, а имена их по большей части говорят сами за себя. DESCRIPTION, HOMEPAGE и LICENSE (используется американская орфография) очевидны, за исключением того, что значение LICENSE должно быть одним из указанных в **/usr/portage/licenses**. SRC_URI – это расположение tar-архива с исходными текстами, $P соответствует имени ebuild'а без расширения **.ebuild** и любых частей **-rN**. Именно поэтому переименование ebuild автоматически заставляется загружать и устанавливать новую версию программы, версия и имя tar-архива берется из имени ebuild. SLOT – обязательный параметр, он обычно равен нулю, что означает, что пакет не множественный [slotted]. Слоты – это способ установки двух, обычно несовместимых, версий одного и того же пакета. Они необходимы, когда различные программы зависят от отличающихся версий. Чтобы увидеть множественные пакеты, выполните

```
equery list --duplicates
```

Equery – еще одна из полезных программ, которую следует установить, она является частью пакета *gentoolkit*. Присутствие IUSE также обязательно, хотя переменная может быть оставлена пустой, в отличие от SLOT; она содержит список флагов USE, соответствующих ebuild. KEYWORDS также не требует пояснений (по крайней мере, если вы читаете этот урок нормально, от начала к концу) и должна иметь значения только тех архитектур, для которых известно, что программа работает. Если вы знаете, что программа не работает на конкретной архитектуре, вам следует предварить ее в KEYWORDS знаком - (минус).

```
KEYWORDS="~x86 amd64 -ppc"
```

означает, что вы протестировали его на трех архитектурах, и он заработал только на первых двух. Любая не указанная архитектура считается не протестированной.

Настраиваем Portage

Каталог **/etc/portage** содержит много файлов, позволяющих перекрыть глобальные настройки для отдельных пакетов напрямую. Чтобы установить флаги USE для выбранных пакетов, используйте **package.use**, тогда как **package.keywords**, как уже говорилось, позволяет изменить для них ACCEPT_KEYWORDS. Если вы хотите установить замаскированный пакет, поместите его в **/etc/portage/package.unmask**; для предотвращения установки указанных вами версий пакетов пригодится **package.mask**.

Здесь можно применять общие или привязанные к версии атомы, хотя будьте осторожны, используя общие атомы в замаскированных и демаскированных файлах.

Чтобы избежать правки файлов **use** и **keyword**, в Gentoo имеется полезный инструмент под названием *flagedit*, способный добавлять или удалять записи из этих файлов. Это особенно полезно при изменении флагов USE для каждого пакета при помощи

```
flagedit net-misc/snafu usb
flagedit net-misc/snafu -usb
flagedit net-misc/snafu %usb
```

Первые два добавляют usb и -usb записи в **package.use**, а последний удаляет любые упоминания о usb (и всю запись snafu, если здесь это последний флаг use), так что будут использоваться стандартные флаги USE. *flagedit* не только существенно экономит время по сравнению с ручным редактированием файлов, но также проверяет, что указанные вами флаги USE верны.

Следующий шаг

Все, чем мы покамест занимались на нашем уроке – это чрезвычайно простой ebuild, даже не рассматривающий фиксированные зависимости, не говоря уж о различных опциях и зависимостях, управляемых флагами USE. Давайте его немного дополним. *Portage* различает зависимости времени компиляции, те, что необходимы только при сборке программы, и зависимости времени исполнения. Они указываются в переменных DEPEND и RDEPEND соответственно, и часто совпадают. Ebuild для Snafu может содержать

```
DEPEND="dev-libs/libusb
  media-libs/libpng
  >=app-misc/unfoo-1.0.5"
RDEPEND="${DEPEND}"
```

Заметьте, что мы можем указать соответствующую версию пакета. А что если *Snafu* поддерживает USB опционально? Ясно, что это встроено в программу, так что с ней уже ничего не сделаешь, но давайте предположим, что в ней имеется опция «не работать с USB-устройствами», и мы хотим использовать эту возможность при ее установке. Именно для этого и существуют флаги USE. Нам необходимо добавить в ebuild три вещи: сообщение, какие флаги USE следует учитывать, учет всех зависимостей, затрагиваемых флагами USE, и пересылку соответствующих опций в конфигурационный скрипт программы, примерно так:

```
IUSE="usb"
DEPEND="usb? (dev-libs/libusb)
  другие зависимости..."
src_compile() {
  econf \
   $(use_enable usb)
  emake || die "Make failed"
}
```

flag? (atom) в DEPEND означает «если флаг установлен, добавить атом к зависимостям». Поскольку мы меняем опции конфигурации, нельзя использовать стандартную функцию src_compile(), которая по существу запускает *econf* для вызова **./configure**, а затем *emake* для запуска *make*; обе они являются некоторыми специфичными для *portage* улучшениями. Мы всего лишь добавляем дополнительные опции в **./configure**. Функция use_enable проверяет, установлен ли флаг USE, и если да, то выводит --enable-flagname. Если в каталоге с исходными текстами *snafu* вы выполните ./configure --help, как при установке из исходных текстов вручную, то увидите список опций. Если опция для включения поддержки USB – это --enable-usb, то добавление $(use_enable usb) в *econf* означает, что **./configure** вызывается с этими опциями. А если вместо этого конфигурационный скрипт использует --enable-libusb? Вы можете определить любую разрешенную строку для использования вместо названий флагов при помощи

```
econf $(use_enable usb libusb)
```

Некоторые конфигурационные скрипты используют --with-option вместо --enable-option; в этом случае замените use_enable на use_with. Если вы emerge'ите пакет при помощи USE="-usb", то *configure* будет запущен с --disable-usb или --without-usb.

Классный код

Базовые функции ebuild, такие как src_compile, доступны для всех ebuild'ов, но существует множество других функций, используемых группами ebuild, которые вы также можете использовать. Они называются eclasses и содержатся в **/usr/portage/eclasses**; их можно добавить в ваш ebuild путем включения

```
inherit имя_класса
```

в самом начале. Наиболее распространена из них eutils, используемая огромным числом ebuild'ов, но также имеются и классы для программ определенных типов, так что если вы пишете ebuild для программы KDE, посмотрите различные kde eclasses. Сами eclasses содержат пояснительные комментарии, но вам не мешает обратиться к **app-portage/portage-manpages** за дополнительной документацией. Другие места, где можно разжиться подробной информацией – man-страницы ebuild и *portage*, а также Руководство разработчика Gentoo на http://devmanual.gentoo.org.

Это лишь кратчайшее введение в ebuild, но вполне достаточное, чтобы вы могли начать, и оно показывает, что написать простой ebuild очень просто, потому что разработчики *portage* уже сделали всю трудную работу за вас. Все равно что в кулинарном рецепте указывать только ингредиенты, будучи счастливым владельцем целого персонала поваров, которые умеют готовить блюдо. LXF

Полезные утилиты

Имеется несколько программ, способных сделать работу с *portage* проще и быстрее. Вот мои любимые, в порядке важности:

» *eix*
Уже описывался: это самый быстрый и гибкий инструмент поиска в *portage*.

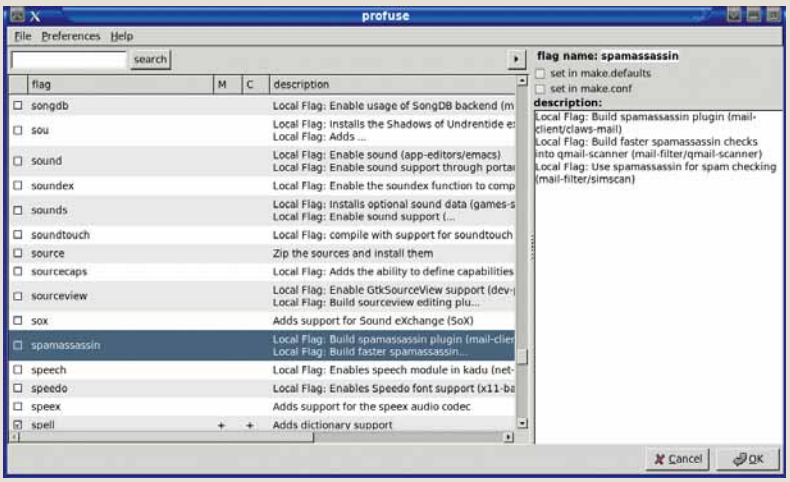
» *gentoolkit & gentoolkit-dev*
Gentoolkit содержит несколько полезных утилит: *equery* для запроса информации о пакетах, *revdep-rebuild* для поддержания системы в порядке, *glsa-check* следует запускать после каждой синхронизации для проверки предупреждений по безопасности, *eclean* может вернуть свободное место путем удаления уже не нужных tar-архивов. *Gentoolkit-dev* содержит еще несколько утилит для работы с ebuild'ами, в основном ориентированных на создание ebuild'ов для распространения.

» *genlop*
Это синтаксический анализатор файла **emerge.log**, не особо замечательный, но он может откопать любую информацию об установленных пакетах и даже предоставить оценку времени до завершения текущего emerge.

» *portage-utils*
Здесь содержатся программы 'Q', выполняющие схожие с *equery* функции. Альтернатива Q работает быстрее, но *equery* имеет больше функций, так что устанавливайте обе.

» *layman*
Уже описывалась; если вы хотите использовать ebuild'ы с оверлеев, *layman* намного упростит управление ими.

» *sufed*, *profuse* и *flagedit*
Это редакторы флагов USE; *ufed* и *profuse* – интерактивные редакторы, работающие с глобальными флагами, отображая информацию о каждом флаге и позволяя включать их или выключать. *Flagedit* – только для командной строки, он подразумевает, что вы уже знаете, чего хотите, но также работает с ключевыми словами [keyword] и с установками для каждого пакета в **/etc/portage**.



› Отображать или менять ваши глобальные флаги USE поможет *Profuse*. Это интерфейс GTK+, но он переключится в *ncurses*, если X недоступен.

» Через месяц Предмет столь захватывающий, что это пока секрет!

Почтовый

ЧАСТЬ 9 Хотите оснастить свою программу возможностью писать «на деревню дедушке»? **Александр Бабаев** знает подходящее средство.

Несмотря на засилье браузера в серии, JEE им не ограничивается. Давайте попробуем посылать письма из Java (ведь каждая хорошая программа должна уметь сообщать разработчикам об ошибках), и сделаем это приложение не браузерным, а «обычным».

Коротко о почте

Чтобы работать с почтой, нужно уметь её отправлять и получать. При попытке разобраться в этом вы наткнетесь на следующие буквосочетания (с разными вариациями): SMTP, POP, IMAP. Рассмотрим кратко, что это такое и как этим пользоваться (а также где почитать поподробнее).

» SMTP

При помощи Simple Mail Transfer Protocol (простого протокола передачи почты) почта передается с клиента на сервер. Отправляется то есть. Протокол текстовый, и если есть желание, можно отправлять письма прямо из telnet'а.

» POP

Тоже текстовый протокол, но уже не для передачи, а для приема сообщений. Может выдать информацию по почтовому ящику (сколько сообщений, какой их размер), загрузить сообщение по номеру и так далее.

» IMAP

POP предполагает, что почта скачивается на клиент и там уже раскладывается по папкам, обрабатывается, группируется. При этом с сервера сообщения стираются. Это не всегда удобно. Как раз для хранения почты на сервере создан протокол IMAP. Как и предыдущие два, он текстовый. Но с его помощью можно не только получить сообщения, но и создать на сервере папку, переместить письмо куда-нибудь, подписаться на получение изменений (новых писем) в папке, и так далее.

» GoogleMail/HotMail/…

Но и это не всё. Протоколы протоколами, но некоторые сервисы работают «по-своему». И если GoogleMail предоставляет POP-интерфейс, то Hotmail, например, нет. В таком случае обычно есть какой-то свой, нестандартный протокол.

Итого

В итоге получается, что разных протоколов много-много (это не считая вариантов и нюансов, комбинаций которых сотни). И чтобы по-человечески все это обрабатывать, пришлось писать бы огромное количество кода. А потом его отлаживать… Поэтому обычно, рассматривая сетевые приложения, отправку/получение писем обходят стороной. Действительно, зачем? Кому нужно, и так разберется.

Но в Java, как всегда, уже позаботились о том, чтобы упростить жизнь человеку, которому нужно рассылать письма – позаботились на самом высшем уровне (в Sun Microsystems) и достаточно качественно.

И где волшебная кнопка?

Есть такая замечательная библиотека, JavaMail. Она достаточно крупная (224 килобайта только JAR-файл), зато и умеет очень много. А чего не умеет – можно научить, благо архитектура настраиваемая. Давайте посмотрим, как с ней работать.

Подготовка

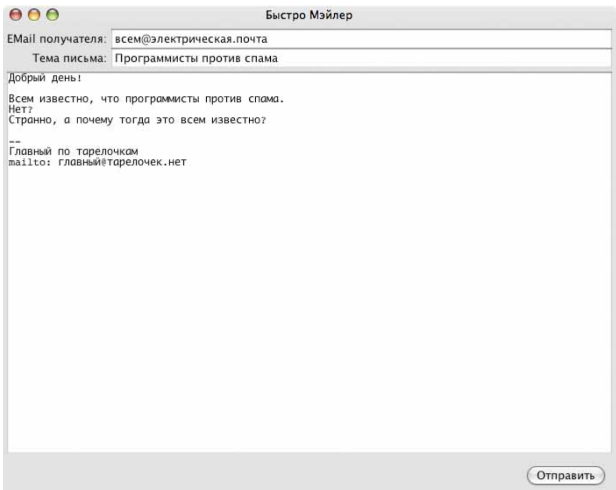
Для начала скачаем саму библиотеку. Страничка продукта находится по адресу http://java.sun.com/products/javamail/; скачивать нужно, как водится, последний релиз (1.4). Также понадобится JavaBeans Activation Framework (JAF), которую можно загрузить со странички рядом: http://java.sun.com/products/javabeans/jaf/index.jsp.

После загрузки и разархивирования, получаем две библиотеки: **mail.jar** и **activation.jar**. Первая из них поддерживает все возможные протоколы, поэтому размер имеет достаточно внушительный. Если что-то из этого многообразия вам не нужно, можно воспользоваться урезанными версиями, они также содержатся в **mail.jar**.

Создадим каталог для проекта (**QuickMailer**), в нем заведем подкаталог **libs** и положим туда эти два jar-файла. Потом заведем другой подкаталог (**src**), для записи исходных текстов.

Окошки

Сделаем окошко для отправки сообщения. Оно будет простое – вот такое:



› **Рис. 1. Окно создания и отправки сообщений.**

» **Месяц назад** Сокеты – основа сетевого взаимодействия и работа с ними из Java.



сервис



Подробно рассказать про то, как создаются формы, не хватит места. Но привести код, создающий такое окошко – запросто.

```
import javax.swing.*;
import javax.swing.border.EmptyBorder;
import java.awt.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class QuickMailerForm extends JFrame {
  private JTextField _fieldTo;
  private JTextField _fieldSubject;
  private JEditorPane _message;
  private JButton _buttonSend;

  public QuickMailerForm() throws HeadlessException {
     setTitle("Быстро Мэйлер");

     setDefaultCloseOperation(EXIT_ON_CLOSE);

     createLayout();
     createActions();

     pack();
     setSize(700, 560);
     Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
     setLocation((int) (screenSize.getWidth() - 700)/2, (int) ((screenSize.
getHeight() - 560)/2));
  }
}
```

Как видно, класс наследуется от JFrame, это окно приложения. Имеется конструктор, где окну присваивается заголовок, устанавливается размер и положение в середине экрана. Также есть две функции: первая создает компоненты (createLayout), вторая «вешает» на кнопку Отправить обработчик события, который собирает информацию и вызывает метод отправки почты.

Вот как создаются компоненты формы:

```
private void createLayout() {
     JPanel labelsPanel = new JPanel(new GridLayout(2, 1));
     labelsPanel.add(new JLabel("EMail получателя:", JLabel.RIGHT));
     labelsPanel.add(new JLabel("Тема письма:", JLabel.RIGHT));

     JPanel buttonsPanel = new JPanel(new FlowLayout(FlowLayout.
RIGHT));
     _buttonSend = new JButton("Отправить");
     buttonsPanel.add(_buttonSend);

     JPanel fieldsPanel = new JPanel(new GridLayout(2, 1));
     _fieldTo = new JTextField();
     _fieldSubject = new JTextField();
     fieldsPanel.add(_fieldTo);
     fieldsPanel.add(_fieldSubject);

     JPanel controlsPanel = new JPanel(new BorderLayout(5, 5));
     controlsPanel.add(labelsPanel, BorderLayout.WEST);
     controlsPanel.add(fieldsPanel, BorderLayout.CENTER);
     JPanel mainPanel = new JPanel(new BorderLayout(5, 5));

     _message = new JEditorPane("text/rtf", "");
     mainPanel.add(controlsPanel, BorderLayout.NORTH);
     mainPanel.add(new JScrollPane(_message), BorderLayout.CENTER);
     mainPanel.add(buttonsPanel, BorderLayout.SOUTH);

     mainPanel.setBorder(new EmptyBorder(5, 5, 5, 5));

     setContentPane(mainPanel);
}
```

Вкратце, здесь создается несколько панелей, вложенных друг в друга. Чтобы выглядело получше, задаются отступы и межкомпонентные расстояния.

Последний метод – создание обработчика события нажатия на кнопку:

```
private void createActions() {
     _buttonSend.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
           try {
              QuickMailer.sendMessage("alex@jdnevnik.com",
_fieldTo.getText(), _fieldSubject.getText(),
                    _message.getDocument().getText(0, _message.
getDocument().getLength()));
           } catch (Exception e1) {
              e1.printStackTrace();
           }
        }
     });
}
```

Тут все достаточно просто. Вытаскиваются параметры письма из полей, после чего вызывается некий метод sendMessage, который мы сейчас и рассмотрим подробнее.

Собственно отправка сообщения

Предполагается, что у вас на localhost'е настроен smtp-сервер (у меня стоит *postfix*), либо есть доступ к какому-то другому (который не требует авторизации: с ней разбираться пока не будем).

Для начала создадим адреса отправителя и получателя:

```
public static void sendMessage(String aFrom, String aTo, String aSubject,
String aMessageText)   throws Exception {
   InternetAddress from = new InternetAddress(aFrom, "From");
   InternetAddress to = new InternetAddress(aTo, "To");
```

Теперь нужно настроить так называемый транспорт, который будет заниматься отправкой сообщения.

```
   Properties props = new Properties();
   props.put("mail.transport.protocol", "smtp");
   props.put("mail.smtp.host", "localhost");
   props.put("mail.smtp.port", "25");
   Session session = Session.getDefaultInstance(props);
   Transport transport = session.getTransport();
```

Теперь – создадим само сообщение.

```
   MimeMessage message = new MimeMessage(session);
   message.setFrom(from);
   message.setRecipient(Message.RecipientType.TO, to);
```

```
    message.setSubject(aSubject, "utf-8");
    message.setContent(aMessageText, "text/plain; charset=utf-8");
```

И, наконец, отошлем письмо.

```
    transport.connect();
    transport.sendMessage(message, new Address[]{to});
    transport.close();
}
```

Как можно заметить, все предельно просто и понятно – но исключительно потому, что сам пример простой. Система достаточно мощная, чтобы справиться и с авторизацией, и с сообщениями на разных языках, и с вложениями файлов.

Собираем все вместе

Осталось только написать метод, который будет все это запускать.

```
public static void main(String[] args) {
    QuickMailerForm form = new QuickMailerForm();
    form.setVisible(true);
}
```

У меня после написания кода получилось два файла, **QuickMailer.java** и **QuickMailerForm.java**. Скомпилируем их (выполнив, находясь в каталоге, в котором находится **src** и **libs**):

```
javac -cp libs/activation.jar:libs/commons-email-1.0.jar:libs/mail.jar
-encoding utf-8 -d out src/*.java
```

Обратите внимание на часть команды после -cp. Это указание компилятору, где искать используемые в коде классы, кроме стандартных. Часть команды после -d определяет каталог, куда будут складываться скомпилированные классы. И, наконец, так как мы создавали файлы в кодировке UTF-8 (в ней представлены исходные тексты на диске), то и в командной строке это нужно указать, иначе будет выбрана кодировка по умолчанию, а это не всегда верно.

Запустим:

```
cd out
java -cp .:../libs/activation.jar:../libs/commons-email-1.0.jar:../libs/mail.jar
QuickMailer
```

Можно отправлять письма.

А получить?

На новое окошко места уже не хватит. Ограничимся просмотром кода, который нужно написать для того, чтобы получить письмо, например, по протоколу POP.

```
Properties props = new Properties();
Session session = Session.getDefaultInstance(props, null);

Store store = session.getStore("pop3");
store.connect(aHost, aUserName, aPassword);

Folder folder = store.getFolder("INBOX");
folder.open(Folder.READ_ONLY);

Message message[] = folder.getMessages();

for (int i = 0, n = message.length; i < n; i++) {
    System.out.println(i + ": " + message[i].getFrom()[0] + "\t" + message[i].
getSubject());
}

folder.close(false);
store.close();
```

Данный кусок кода просто выведет список всех писем на сервере.



Ну, а если не мудрить…

Есть вариант и попроще. Если программа работает с почтой активно, можно использовать библиотеку наших постоянных друзей из apache-commons. Называется она *commons-email*, и ее страничка располагается по адресу http://commons.apache.org/email/. Скачав библиотеку, положим её в **libs**, к **mail.jar** и компании. Теперь попробуем отправить письмо с ее помощью:

```
public static void sendMessageCommonsEMail(String aFrom, String aTo,
String aSubject, String aMessageText) throws EmailException {
    SimpleEmail email = new SimpleEmail();
    email.setHostName("localhost");
    email.setFrom(aFrom, "From");
    email.addTo(aTo, "To");
    email.setSubject(aSubject);
    email.setMsg(aMessageText);
    email.send();
}
```

Насколько все проще и понятнее сразу стало! А если нужно файл приложить? Да пожалуйста:

```
public static void sendMessageWithAttachment(String aFrom, String aTo,
String aSubject, String aMessageText) throws EmailException {
    EmailAttachment attachment = new EmailAttachment();
    attachment.setPath("attachments/attachment.zip");
    attachment.setDisposition(EmailAttachment.ATTACHMENT);
    attachment.setDescription("Файл-приложение к письму");
    attachment.setName("attachment.zip");

    MultiPartEmail email = new MultiPartEmail();
    email.setHostName("localhost");
    email.setFrom(aFrom, "From");
    email.addTo(aTo, "To");
    email.setSubject(aSubject);
    email.setMsg(aMessageText);

    email.attach(attachment);

    email.send();
}
```

Здесь тоже ничего сложного нет. Создать приложение в письме можно и используя только JavaMail, но там это получается достаточно непросто, и длиннее раза в три-четыре.

Про спам

Конечно, работа с почтой не так проста, как это отражено в статье. Есть и проблема спама (а для программно отправляемых сообщений – проблема того, что оно с большой вероятностью посчитается именно спамом), и проблема корректности. SMTP-протокол, в частности, достаточно старый, и там много неточностей, неявных правил и так далее. В общем, то, что есть JavaMail – это отлично, и она очень сильно помогает при работе с почтовыми сообщениями, но панацеей тем не менее не является. Все равно нужно представлять себе, как работает протокол, какие заголовки нужно ставить, как обрабатывают сообщения разные почтовые клиенты (чтобы письмо нормально там показывалось, а не крякозябрами) и много чего еще.

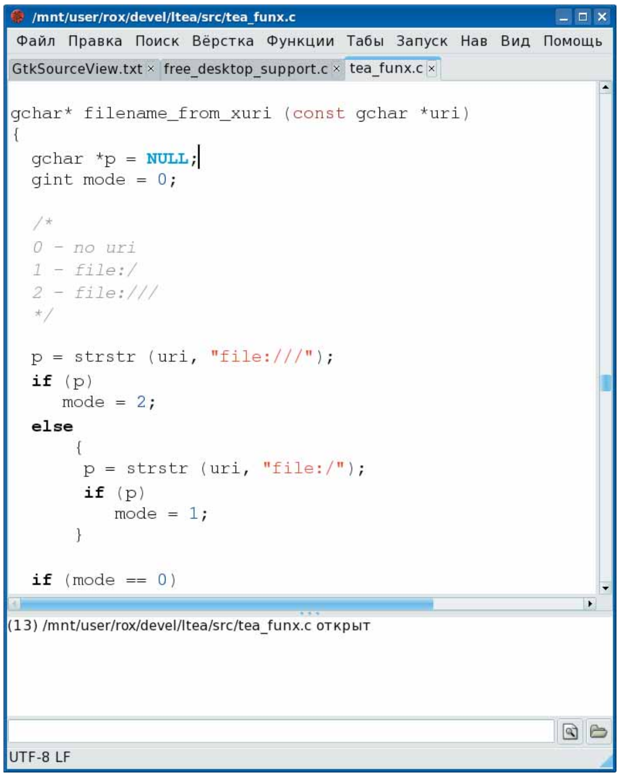
Но все же, надеюсь, теперь можно не бояться страшных буквенных сочетаний, связанных с почтой, и спокойно встраивать в программу еще одно удобнейшее средство коммуникации: электронные письма. LXF

» Через месяц Struts: полезная библиотека, в которой собрано сразу несколько интересных решений.

Движок текстового редактора

Если возможностей чистого *GtkTextView* для вашего приложения уже недостаточно или вы просто решили по-быстрому написать лучшую IDE всех времен и народов – начните с урока *GtkSourceView* от **Петра Семилетова**.

В состав библиотеки *Gtk* входит отличный виджет текстового редактора, воплощенный в виде взаимодополняющих классов GtkTextBuffer, GtkSourceView и ряда других, вспомогательных (мы писали про них в LXF93). Но ради универсальности этого виджета его разработчики пожертвовали некоторыми вещами. Например, нет встроенного механизма redo/undo, нет также подсветки синтаксиса различных языков программирования, хотя виджет обладает способностью задавать цвет и параметры шрифта при отображении отдельных участков текста.



› Рис. 1. Подсветка в редакторе *TEA*, выполненная с помощью GtkSourceView 2.

Однако среди разработчиков известного редактора *GEdit* вырос виджет-надстройка над *GetTextView* – и называется он *GtkSourceView* (**gtksourceview.sourceforge.net**). В нем есть все то, чего недостает в стандартном виджете: и движок redo/undo, и подсветка синтаксиса, и отображение правой границы – так называемого margin'a, и многое другое.

Сейчас развитие *GtkSourceView* стоит на пороге перехода на новую ветку – под номером два. Вторая версия еще считается нестабильной, а API ее не устоялось. В этой статье я постараюсь дать представление и о современной версии библиотеки, и о грядущей. Замечу также, что я предполагаю ваше умение работать с обычным *GtkTextView*. Как я упоминал, *GtkSourceView* – лишь надстройка над *GtkTextView*, поэтому работа с *GtkSourceView* подразумевает использование и «стандартных» функций из *Gtk*.

Чтобы создать новый виджет *GtkSourceView*, используется функция Gtk_source_view_new:

```
GtkWidget *text_view = gtk_source_view_new ();
```

Классу буфера для *GtkSourceView*, вместо GtkTextBuffer, соответствует GtkSourceBuffer. Для его создания надо вызвать функцию gtk_source_buffer_new:

```
GtkSourceBuffer *text_buffer = gtk_source_buffer_new (table);
```

В качестве параметра мы передаем этой функции экземпляр таблицы с определениями тэгов пользовательской подсветки (вне движка подсветки *GtkSourceView*). Напомню, что в текстовом движке *Gtk*, тэг – это объект, содержащий параметры форматирования текста, как то: шрифт, цвет и тому подобное.

Для движка *GtkSourceView* первой версии, таблица имеет тип GtkSourceTagTable. Во второй версии *GtkSourceView* надо использовать стандартный тип из *Gtk* – GtkTextTagTable. Можно передавать в параметре значение NULL. Тэги вам понадобятся, только если вы захотите неким образом размечать текст – например, подчеркивать слова с ошибками. Создание тэгов, помещение их в таблицу и последующее применение (тэги применяются в тексте к месту, ограниченному итераторами) – тема, выходящая за рамки данной статьи.

Для получения текста из GtkSourceBuffer и помещения его туда применяются те же функции, что и для GtkTextBuffer – gtk_text_buffer_get_text и gtk_text_buffer_set_text. Передавая экземпляр GtkSourceBuffer в качестве параметра, можете приводить его к типу (макросом GTK_TEXT_BUFFER), а можете не приводить – все равно передается указатель, а компилятор может выдавать предупрежде- »

ния в зависимости от своих настроек. Однако ничего дурного не случится, если вы вместо:

```
gchar *text = gtk_text_buffer_get_text (GTK_TEXT_BUFFER (buffer));
```

напишете такое:

```
gchar *text = gtk_text_buffer_get_text (buffer);
```

В общем, здесь и далее, ради экономии журнального места, я не буду использовать в примерах приведение к типу.

Итак, с текстовым содержимым GtkSourceBuffer мы разобрались. Теперь давайте применим к загруженному тексту подсветку синтаксиса – если таковая, конечно же, предусмотрена для конкретного языка движком: создания подсветки «пользовательского» языка мы касаться не будем. Используем готовые, «встроенные» модули подсветки – физически это XML-файлы, в которых заданы правила подсветки.

Само собой, виджет не догадывается по помещенному в него тексту, подсветку синтаксиса какого языка надо использовать. Эту процедуру необходимо запрограммировать Работать будем опять-таки с буфером.

В *GtkSourceView 1*:

```
gtk_source_buffer_set_language (buffer, language);
gtk_source_buffer_set_highlight (buffer, TRUE);
```

В *GtkSourceView 2*:

```
gtk_source_buffer_set_language (buffer, language);
gtk_source_buffer_set_highlight_syntax (buffer, TRUE);
```

Функция gtk_source_buffer_set_language устанавливает нужную подсветку (удаляя при этом все тэги типа *Gtk*SourceTag из таблицы тэгов переданного буфера), а gtk_source_buffer_set_highlight (либо gtk_source_buffer_set_highlight_syntax) включает ее. А вот что такое переменная language в параметре первой функции? Определена она так:

```
GtkSourceLanguage *language = NULL;
```

Каким образом связать переменную language с подсветкой файлов определенного типа? В первой версии *GtkSourceView* есть функция с очень длинным названием: gtk_source_languages_manager_get_language_from_mime_type. Она получает два параметра – указатель на менеджер языков и строку, задающую тип MIME (для загруженного файла).

Менеджер языков – это переменная типа GtkSourceLanguagesManager (в *GtkSourceView 1*) и GtkSourceLanguageManager (в *GtkSourceView 2*). В первом случае получаем экземпляр нужного класса с помощью функции gtk_source_languages_manager_new, во втором лучше использовать новое средство, функцию gtk_source_language_manager_get_default. Вручную уничтожать полученные объекты не нужно.

Итак, для получения механизма подсветки, соответствующего некоторому типу MIME, используется примерно следующее:

```
language = gtk_source_languages_manager_get_language_from_mime_type (source_languages_manager, mime_string);
```

Однако тут возникают два вопроса. Как быть в *GtkSourceView 2*, где покамест нет функции, «отдающей» нам язык по переданному типу MIME? И как вообще получить MIME для файла?

На первый вопрос ответ очевиден – надо писать свою функцию. В каталоге tests исходных текстов *GtkSourceView 2* есть образец – смотрите файл **test-widget.c**, функцию get_language_for_mime_type. Не самое удачное воплощение алгоритма, но оно работает, и вы будете иметь представление, куда двигаться дальше.

Теперь о типах MIME. Есть несколько библиотек, позволяющих работать с ними. Самый естественный способ для программы, основанной на *Gtk* – обратиться к библиотеке GnomeVFS (LXF94):

```
GnomeVFSFileInfo *info = gnome_vfs_file_info_new ();
GnomeVFSResult r = gnome_vfs_get_file_info (filename, info, GNOME_VFS_FILE_INFO_GET_MIME_TYPE);

gchar *mime = gnome_vfs_file_info_get_mime_type (info);

// тут что-то делаем со строкой mime, затем освобождаем память, выделенную под структуру info:

gnome_vfs_file_info_unref (info);
```

Итого – в переменной mime, покуда память для info не освобождена, имеем значение, соответствующее типу MIME для файла, чье имя передано в переменной filename.

Не забудьте, что перед использованием функций **GnomeVFS** надо инициализировать эту библиотеку (gnome_vfs_init), а после завершения работы вызвать gnome_vfs_shutdown.

Итак, подсветка заработала. Что еще мы можем с ней сделать? Изменить цвета. В *GtkSourceView 1* можно, с помощью довольно громоздкого кода, назначить параметры подсветки для каждого элемента каждого доступного языка, например, для строки в HTML или CSS. В *GtkSourceView 2* появился более удобный способ изменения цветов – использование схем. Схема здесь – это, грубо говоря, предустановленный набор цветов. Во время написания этих строк, в составе *GtkSourceView* идет 3 схемы – classic, kate и tango. Как выбрать схему?

Сначала раздобудем список идентификаторов установленных в системе схем:

```
GtkSourceStyleSchemeManager *manager = gtk_source_style_scheme_
manager_get_default ();

gchar **scheme_ids;
g_object_get (manager, "scheme-ids", &scheme_ids, NULL);

GList *ids = NULL;
gint c = g_strv_length (scheme_ids) - 1;
gint i;

for (i = 0; i <= c; i++)
   ids = g_list_prepend (ids, scheme_ids[i]);
```

Здесь мы получаем менеджер схем по умолчанию, а затем считываем из него свойство scheme-ids в массив scheme_ids. После чего, для удобства, перемещаем идентификаторы в список ids. Со списком GList удобнее работать, если у вас есть готовые функции для создания виджетов-меню или виджетов-списков на основе данных из GList. А массивы представляются мне чем-то старорежимным.

Далее, зная идентификаторы доступных схем, мы можем затребовать нужную схему по ее идентификатору:

```
GtkSourceStyleScheme *scheme = gtk_source_style_scheme_manager_
get_scheme  (manager, cheme_id);
```

Осталось лишь назначить схему буферу *Gtk*SourceBuffer:

```
gtk_source_buffer_set_style_scheme (buffer, scheme);
```

Чем еще полезным нас может порадовать *GtkSourceView2*? Механизмом Undo/Redo. По умолчанию любое изменение текста в буфере подлежит отмене либо обратному действию. Если вам надо временно этот механизм отключить, используйте функции gtk_source_buffer_begin_not_undoable_action и gtk_source_buffer_end_not_undoable_action. Любые действия с текстом в буфере между их вызовами не будут включены в очередь отмены. Так, например, следует поступить при первой вставке текста из загруженного файла – иначе, когда пользователь будет применять отмену, он рано или поздно дойдет до пустого документа.

Для программного вызова отмены и отмены отмены (redo, как вы догадались) нам даны функции gtk_source_buffer_undo **и** gtk_source_buffer_redo. Можно также проверить, доступны ли эти операции, и установить наибольшую длину очереди движка замен – указать, сколько в ней может быть элементов.

Кроме того, движок *GtkSourceView* предоставляет много функций, полезных разработчикам редакторов для программистов. Это управление автоматическим отступом и параметрами табуляции, отображение номеров строк слева от текста, работа с маркерами (полезно для показа каких-нибудь закладок) и многое другое.

Обе версии *GtkSourceView* – первую и вторую – можно установить одновременно, они не мешают одна другой. Если вы пишете программу с использованием *GtkSourceView*, то стоит подумать над поддержкой обеих версий, ведь в старых дистрибутивах (да и в стабильной ветке Debian) вторая версия появится не скоро.

Проверить наличие установленной библиотеки и включить ее в параметры сборки можно разными способами. Приведенный ниже код (предназначенный для размещения в **configure.in**) проверяет, установлены ли в системы библиотеки *GtkSourceView2* и *GnomeVFS 2*. Если да, то они добавляются к параметрам сборки, а в файле **config.h** будет определена константа gtksourceview2_SUPPORTED, проверяя которую, мы сможем узнать, есть поддержка *GtkSourceView2* или нет:

```
echo -n "checking for GtkSourceView2... "

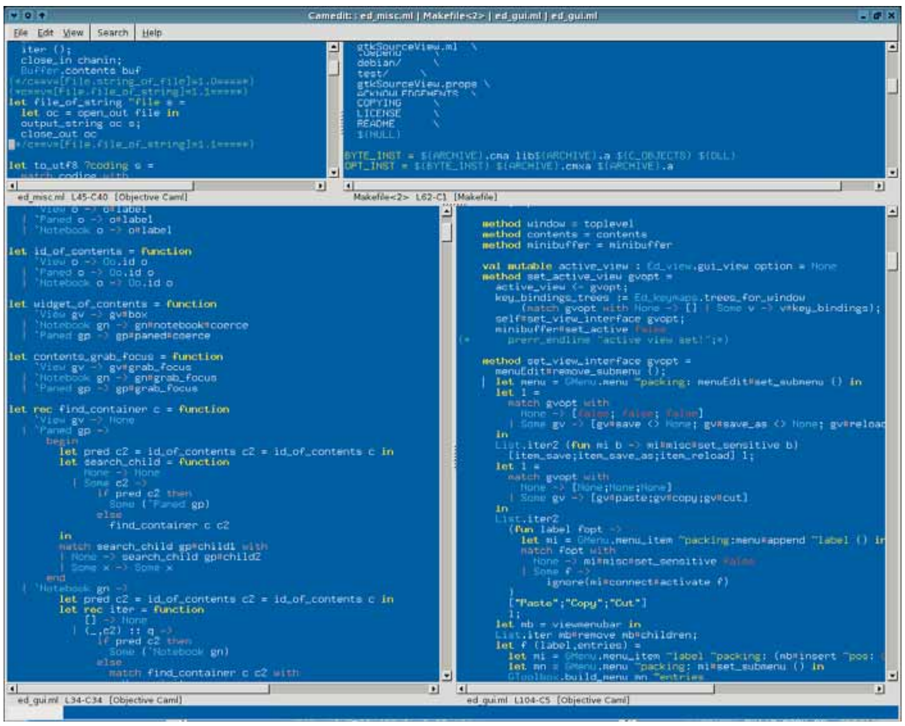
if pkg-config --exists gtksourceview-2.0  ; then
          LIBS="$LIBS `pkg-config --libs gtksourceview-2.0 `"
          CFLAGS="$CFLAGS `pkg-config --cflags gtksourceview-2.0 `"
          AC_DEFINE(GTKSOURCEVIEW2_SUPPORTED, 1,
[GTKSOURCEVIEW2_SUPPORTED])
          echo "yes"

          echo -n "checking for gnome-vfs-2.0... "
          if pkg-config --exists gnome-vfs-2.0 ; then
                    LIBS="$LIBS `pkg-config --libs gnome-vfs-2.0`"
                    CFLAGS="$CFLAGS `pkg-config --cflags gnome-
vfs-2.0`"
                    echo "yes"
          else
                    echo "no"
          fi

else
          echo "no"
fi
```

Сходным способом можно проверить и наличие *GtkSourceView 1*.

Все это касается стандартной системы сборки/установки с помощью *Autotools* – *Scons*, *CMake* и прочие настраиваются аналогично (с поправкой на синтаксис, разумеется). Для удобной работы с пользовательской настройкой подсветки в первой версии *GtkSourceView* вам еще понадобится библиотека *GConf*, чтобы держать настройки цветов в хранилище. Все подробности можно найти в исходных текстах *GEdit* или *TEA*. LXF



Новая серия! Изучаем учетную платформу со свободной лицензией

Первый взгляд

ЧАСТЬ 1 Андрей Паскаль научит отличать спелый Ананас не только по виду и запаху, покажет несколько приемов препарирования чудо-фрукта и... даст пару советов по уходу за офисными Ананасами.



Наш эксперт

Андрей Паскаль использует Linux более семи лет и играет роль координатора проекта Ананас. Он также знает ответ на популярный у всех новичков вопрос: «Какой дистрибутив Linux мне выбрать?».

За забавным названием «Ананас» скрывается – «кто бы мог подумать!» – полноценная среда разработки и исполнения со своим языком программирования, редактором экранных диалогов, шаблонами печатных форм, метаданными и удобным редактором структур данных. Добавим сюда архитектуру «клиент-сервер» с поддержкой наиболее популярных свободных SQL-серверов *MySQL*, *PostgreSQL*, умение выживать в среде Linux и Windows и получим (ого!) свободную платформу для автоматизации бизнес-процессов.

Итак, обо всем по порядку. Да, действительно, *Ананас* – это не просто еще одна программа для печати платежных поручений: *Ананас* – первая учетная платформа для Linux. Кроме того, приложения, разработанные на платформе *Ананас*, одинаково хорошо работают и под Linux, и под Windows.

Чем же *Ананас* отличается от хорошо всем известных интегрированных сред программирования, таких как *NetBeans*, *Eclipse* или *Borland Kylix*? – спросит прозорливый читатель. Неужели *Ананас* лучше, или же всерьез собирается конкурировать с известными универсальными инструментами? Разумеется, нет.

Ананас является специализированной платформой, средой разработки учетных бизнес-приложений. Отсюда и вытекают отличия, заложенные при проектировании. Его не следует использовать для написания почтовых серверов или драйверов, управляющих оборудованием, если только вам не доставляет удовольствие занятие, напоминающее вычерпывание супа шумовкой.

Если при программировании универсальным инструментарием программист живет в поле таких понятий, как Класс, Объект, Запись, Поле реляционной таблицы, то программирование со вкусом *Ананаса* составлено из букета таких понятий, как Документ, Справочник, Регистр, Журнал, Отчет.

Для чего это сделано? А для того, чтобы программист мог проще находить общий язык с заказчиком, который живет примерно в том же понятийном мире. Для того, чтобы при составлении программы программист оперировал как можно более высокоуровневыми понятиями и достигал результата как можно с меньшими затратами, не заглядывая внутрь без особой необходимости.

Ну и хватит умствований. Давайте просто посмотрим, из чего же состоит приложение, разработанное на платформе *Ананас*. Но сначала решим задачу установки *Ананаса*.

В свое время я был поражен, узнав, что ананасы произрастают в траве, как арбузы, а не на пальмах, как финики. Поэтому давайте правильно сориентируемся на местности и определим, где же взять *Ананас*. Так как нас интересуют только спелые, готовые к употреблению *Ананасы*, придется проявить выдержку и смекалку при поиске, отборе и тестировании...

Первым местом, которое следует посетить всякому уважающему себя детективу – пардон, опытному пользователю – является, если не считать DVD-диск с вашим любимым дистрибутивом Linux (а также DVD текущего номера *Linux Format*), сайт проекта *Ананас* http://ananas.lrn.ru. Там вам скажут, где лежит продукция цеха по упаковке спелых *Ананасов* в формат, установленный таможенным кодексом вашего государства, или, другими словами, менеджера пакетов вашего дистрибутива.

По обнаружении подходящих пакетов вам не потребуется много знаний для их установки. Ниже я расскажу, как быть, если подходящих к вашему дистрибутиву Linux пакетов нет, или вы не тот человек, который, имея доступ к свежему коду в репозитории CVS, станет довольствоваться упаковкой с датой изготовления полугодичной давности.

Отправляясь в волшебное путешествие, где вас ждут трудности и испытания, помните одно: вам всегда помогут на форуме проекта. Стоит лишь правильно задать вопрос, и вы тут же получите на него исчерпывающий ответ. Форум лежит в глухом, заброшенном местечке с неказистым адресом http://ananas.lrn.ru/phpBB2/.

Ананас из исходников

Чтобы собрать *Ананас* из исходных текстов, вам потребуются следующие обязательные ингредиенты: mysql-devel, qt-devel, qt-qsa-devel. Названия пакетов могут меняться в зависимости от дистрибутива. Приведенные выше именования пакетов и ниже – названия файловых путей приняты в Fedora 7.

После этого – тянем-потянем *Ананас*:

```
cvs -d :pserver:anonymous@www.leaderit.ru:/var/cvs login
```

Система запросит пароль. В ответ нажмите Enter.

```
cvs -d :pserver:anonymous@www.leaderit.ru:/var/cvs co -r release-094b ananas
```

В последней команде стоит обратить внимание на опцию -r release-094b. Она задает версию *Ананаса*, которая будет вытянута в результате выполнения команды.

Возможно, ко времени публикации статьи выйдет более свежая версия; тогда опцию следует сменить на release-XYZb, где X,Y и Z соответствуют номеру версии *Ананаса*, которая указывается в формате ananas-X.Y.Z.

Самые смелые могут и вовсе не указывать эту опцию, чтобы получить исходный код *Ананаса* с самыми свежими изменениями, которые разработчики *Ананаса* поместили в репозиторий вчера или даже сегодня (главное, чтобы не в момент запуска вами команды!):

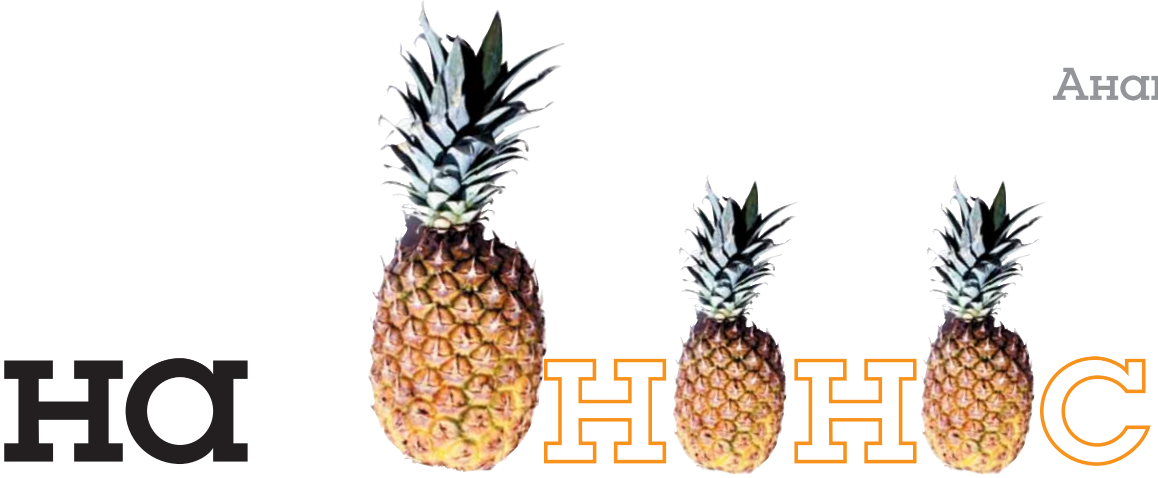
```
cvs -d :pserver:anonymous@www.leaderit.ru:/var/cvs co ananas
```

Репозиторий проекта поддерживается разработчиками в состоянии постоянной готовности к сборке. Это гарантирует успешную компиляцию полученного кода, но не его работоспособность.

Следующая пара команд запустит процесс сборки *Ананаса*.

```
cd ananas
make
```

После успешного завершения процедуры потребуется всыпать в консоль вагон и маленькую тележку команд для правильной установки *Ананаса* в систему. Радует лишь то, что cut-n-paste еще никто не отменял. Разумеется, тут не обойтись без прав пользователя root.



Сначала установим файлы в систему:

```
sudo cp src/ananas/ananas /usr/bin/
sudo cp src/designer/ananas-designer /usr/bin/
sudo cp -P src/lib/libananas.so* /usr/lib/
sudo cp src/plugins/libananasplugin.so /usr/lib/
sudo ln -s /usr/lib/libananasplugin.so /usr/lib/qt-3.3/plugins/designer/
sudo mkdir -p /usr/share/ananas/translations
/usr/lib/qt-3.3/bin/lrelease translations/*.ts
sudo cp translations/*.qm /usr/share/ananas/translations/
sudo cp -r src/designer/templates /usr/share/ananas/
sudo mkdir -p /usr/share/ananas/inventory
sudo cp applications/inventory/*.odt /usr/share/ananas/inventory/
sudo cp applications/inventory/*.ods /usr/share/ananas/inventory/
sudo mkdir -p /usr/share/ananas/extensions
sudo cp src/extensions/libaext_meta.so /usr/share/ananas/extensions/
sudo cp src/extensions/libaextxml.so /usr/share/ananas/extensions/
sudo cp /src/extensions/libaexttext.so /usr/share/ananas/extensions/
sudo mkdir -p /etc/ananas
sudo cp applications/inventory/inventory.cfg /etc/ananas/
sudo cp applications/inventory/inventory.rc /etc/ananas/
sudo cp applications/inventory/inventory-demo.cfg /etc/ananas/
sudo cp applications/inventory/inventory-demo.rc /etc/ananas/
```

Затем создадим и наполним демонстрационную базу данных на *MySQL*-сервере, подразумевая, что вирус паранойи информационной безопасности еще не добрался до вас и вы не меняли пароль пользователю root сервера *MySQL* после установки.

```
mysqladmin -u root create ananas_system
mysqladmin -u root create ananas_inventory
mysqladmin -u root create ananas_inventory_demo
mysql -u root ananas_inventory < applications/inventory/inventory.sql
mysql -u root ananas_inventory_demo < applications/inventory/inventory-demo.sql
```

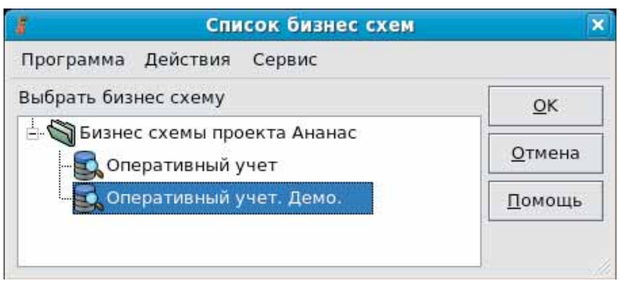
На этом процедуру установки можно считать завершенной. Попробуем попрактиковаться – к счастью, создатели *Ананаса* предусмотрели для вас кое-какие демо-приложения. Для работы с приложением «Оперативный учет» следует запустить *Ананас* командой

```
ananas
```

и выбрать бизнес-схему с соответствующим названием.

Знакомство с интерфейсом

Если программа была правильно установлена, на экране появится окно следующего вида (**Рис. 1**):



› Рис. 1. Диалог выбора бизнес-схемы при запуске программы.

Внешний вид и содержание окна может отличаться от приведенного здесь примера в зависимости от версии и настройки программы, настроек вашей графической среды, но в заголовке (шапке) окна обязательно будет надпись Список бизнес-схем, а справа кнопки, показанные на снимке экрана.

Бизнес-схема

Так называют приложение, разработанное на платформе *Ананас*, потому что помимо собственно кода приложения, сюда включается описание структур данных приложения (метаданные) и база данных с хранящейся в ней информацией. Жизненный цикл бизнес-схем *Ананаса* также несколько отличен от жизненного цикла классических приложений/программ. Бизнес-схемы *Ананаса* рассчитаны на модификацию во время эксплуатации, без потери уже занесенных данных. Таковы требования реального бизнеса в реальной жизни.

Вообще разработчики любят использовать слово «бизнес» в своей терминологии. Работая с *Ананасом*, вы также столкнетесь с такими понятиями, как бизнес-объект, бизнес-процесс.

В списке в средней части окна будут две строки с надписями: «Оперативный учет» и «Оперативный учет. Демо.».

Бизнес-схема «Оперативный учет. Демо.» содержит демонстрационную базу данных. Ее мы и выберем (**Рис. 2**), чтобы ознакомится с возможностями работы приложения и увидеть примеры заполнения справочников и документов. Бизнес-схема «Оперативный учет» предназначена для начала реальной работы и не содержит никакой информации в базе данных, а также и в справочниках.

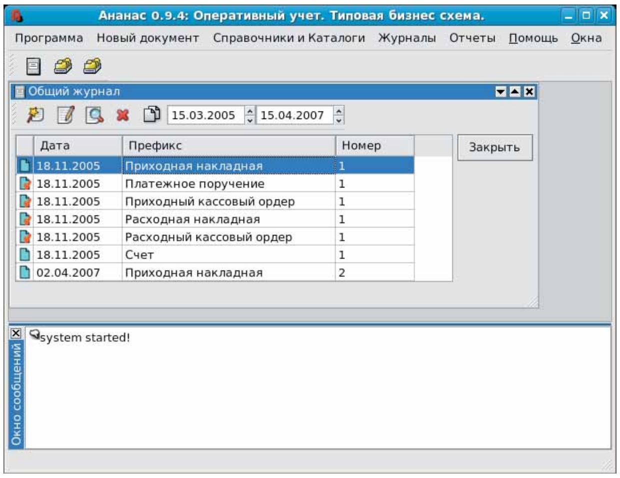
Каждая бизнес-схема использует свою базу данных и является полностью автономной и не связанной с другой бизнес-схемой. То есть, если, работая с демонстрационной бизнес-схемой, вы занесете в справочник продавцов новую организацию, она не появится в справочнике продавцов рабочей бизнес-схемы.

Таких бизнес-схем может быть несколько, что позволяет вести учет при помощи одной программы сразу для нескольких субъектов или разбивать учет на периоды (например, годичные) и для каждого периода выделять отдельную бизнес-схему.

После выбора бизнес-схемы на экране появится основное окно программы, из которого осуществляется доступ ко всем возможностям приложения.

Какие бывают бизнес-схемы?

Помимо бизнес-схемы, поставляемой с дистрибутивом *Ананаса*, существует ряд бизнес-схем, разрабатываемых и поддерживаемых независимыми разработчиками. На сегодня в качестве примера можно привести такие бизнес-схемы, как «Учет транспортных услуг (логистика)», «Оперативный учет в оптовой торговле», «Спутник. Учет заявок». Количество и разнообразие бизнес-схем растет по мере освоения программирования на *Ананасе* новыми разработчиками. Разумеется, не все бизнес-схемы доступны (или будут доступны) свободно. Тем не менее, открытых бизнес-схем всегда будет достаточно для повседневного извлечения вкусной пользы из *Ананаса*.

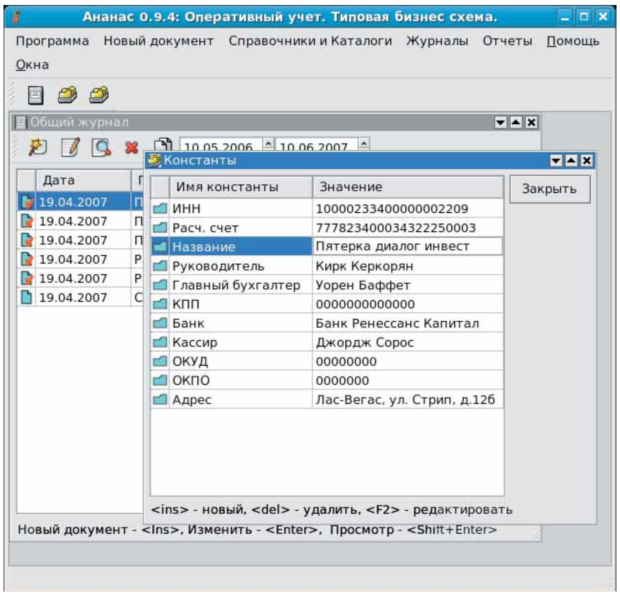


› Рис. 2. Главное окно приложения «Оперативный учет. Демо» после первого запуска.

Бизнес-схема «Оперативный учет. Демо.» достаточно проста в освоении, чему, несомненно, способствуют примеры введенных документов и заполненные справочники. Кроме того, разработчики позаботились и о пользователях-новичках, снабдив приложение подробным 50-страничным руководством, где по шагам и экранным снимкам разбирается каждая возможность бизнес-схемы «Оперативный учет».

Повседневную, будничную работу пользователя бизнес-схемы «Оперативный учет» *Ананаса* по большому счету можно разделить на два вида деятельности:

- Оформление документов, которые называются в бухгалтерском учете первичными учетными документами. Оформление подразумевает создание новых электронных документов, их заполнение и распечатку. Здесь можно назвать такие документы, как накладные на товар, кассовые ордера, счет, банковские поручения.
- Подготовку отчетов/печатных документов для налоговой инспекции или для решения собственных задач по анализу товарных запасов, денежных расчетов, задолженностей и прочих показателей бизнеса.



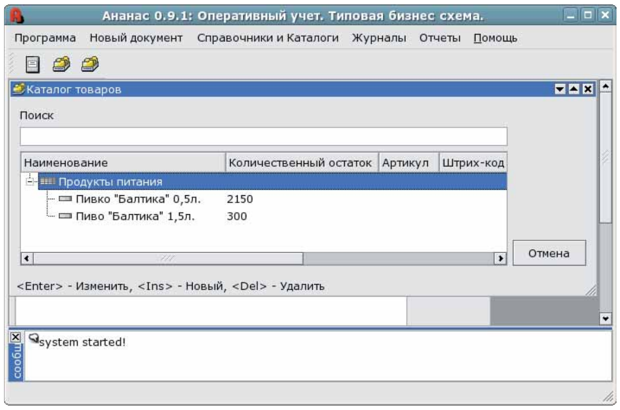
› Рис. 3. Работа со справочником констант.

Разумеется, присутствуют и разовые, эпизодические работы, необходимые для поддержки справочника Контрагентов (продавцов, покупателей) и Каталога товаров в актуальном состоянии.

Наименьшее количество хлопот доставляет справочник Константы (**Рис. 3**): будучи заполненным первым в самом начале эксплуатации *Ананаса*, он практически не меняется со временем. Справочник Константы хранит такую редко меняющуюся информацию, как реквизиты вашей организации, ФИО директора, бухгалтера или просто ваши данные, если вы предприниматель.

Каталог товаров

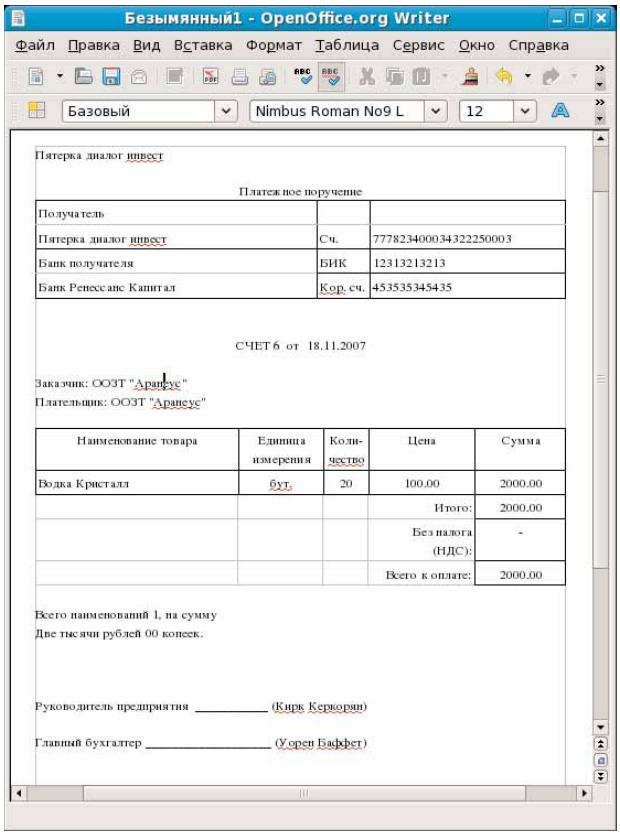
Для хранения информации о товарах в бизнес-схеме «Оперативный учет» используется многоуровневый Каталог товаров (**Рис. 4**). По каждому товару в каталоге отражается состояние остатков в количественном выражении по состоянию на текущий момент времени.



› Рис. 4. В Каталоге товаров отображается количественный остаток по состоянию на текущий момент времени.

Остаток меняется автоматически по мере оформления приходных и расходных накладных, свидетельствующих соответственно о покупке и продаже товаров, в них указанных.

Таким образом, пользователь бизнес-схемы «Оперативный учет» *Ананаса* в любой момент времени имеет достоверную информацию



› Рис. 5. Счет на приобретение товара, сформированный для и открытый в *OpenOffice.org* Ананасом.

о состоянии запасов товаров и может своевременно спланировать их пополнение.

Печать документов и отчетов

Платформа *Ананас* предоставляет программисту возможность создания отчетов в формате *OpenOffice.org*, HTML или любом другом формате документов, хранимых в виде ASCII-файла.

Разработчики бизнес-схемы «Оперативный учет» сделали ставку на использование *OpenOffice.org* как на наиболее удобный для пользователя пакет подготовки документов для печати (**Рис. 5**).

Впечатляет удобство получения документов *OpenOffice.org* из *Ананаса*. Стоит пользователю нажать кнопку «Печать» в экранной форме любого из электронных документов, перед ним тут же появляется окно *OpenOffice.org* с документом, готовым к выводу на печать, а если нужно, и редактированию или конвертации в другие форматы.

В заключение, для краткого перечисления возможностей бизнес-схемы «Оперативный учет», процитирую «Руководство пользователя» этой бизнес-схемы, которое содержит исчерпывающее описание всех ее возможностей.

Бизнес-схема «Оперативный учет» обеспечивает следующую функциональность:

» Ведение ежедневно необходимого учета: покупка, продажа, платежи через банк, кассовые операции;
» Получение управленческой информации: по товарным запасам в суммовом и количественном выражении;
» Получение баланса по расчетам с партнерами;
» Автоматический расчет остатков;
» Генерация отчетов и печать документов;
» Сервисные функции по резервному копированию и восстановлению данных.

Остается пожелать удачи всем, кто, прочитав статью, решил познакомиться с *Ананасом* самостоятельно. А для тех, кто решил не останавливаться на уровне пользователя и готов взяться за программирование, из которых вы узнаете кое-что о CRUD-приложениях, меню, регистрах и отчетах. LXF

Состав объектов бизнес-схемы «Оперативный учет»:

» Справочники
- Справочник контрагентов, отображающий информацию о расчетном балансе каждого контрагента
- Справочник товаров, отображающий информацию по остаткам товаров в количественном и денежном выражении на складе

» Вспомогательные справочники
- Справочник Складов
- Справочник реквизитов (констант)

» Документы
- Счет
- Приходная накладная
- Расходная накладная (ручной ввод или автоформирование по Счету)
- Платежное поручение
- Банковская выписка
- Приходный кассовый ордер
- Расходный кассовый ордер

» Журналы
- Общий

» Специальные журналы
- Журнал реализации
- Журнал прихода

» Отчеты
- Движение товаров
- Ведомость по поставщикам и покупателям
- Ведомость по кассе
- Книга покупок
- Книга продаж

» Печатные формы
- Печатная форма расходной накладной
- Печатная форма приходной накладной
- Печатная форма платежного поручения
- Печатная форма банковской выписки
- Печатная форма ПКО
- Печатная форма РКО
- Печатная форма счета
- Печатная форма счета-фактуры



Фото: http//www.free-pictures-photos.com ©

» **Через месяц** CRUD-приложение в терминах Ананаса: что это значит и как это сделать.



Пингвин впадает в спячку

Приходится часто включать и выключать ноутбук? **Сергей Яремчук** подскажет, как сэкономить пару-тройку минут на загрузке и запуске необходимых приложений.

К сожалению, праздники и выходные имеют вредную привычку быстро заканчиваться, после чего снова предстоит приступать к работе, вспоминая, чем занимался в последний раз. Очевидно, это настолько существенная проблема для народа, что в Windows, начиная с XP, уже реализована возможность гибернации (hibernate state), доступная через **Панель Управления: Производительность и обслуживание > Электропитание > Спящий режим > Разрешить использование спящего режима**. При этом все данные, находящиеся в оперативной памяти, сбрасываются на жесткий диск (т.е. фактически делается моментальный снимок состояния системы), и компьютер отключается. При включении полностью восстанавливается предыдущее состояние, и пользователь начинает работу именно с того места, где он закончил ее. Кроме очевидного удобства, выключение/включение компьютера при использовании такого режима ускоряется, так как не происходит закрытия и запуска приложений, что очень важно для мобильных устройств, нацеленных на энергосбережение.

А что пингвины?

Сторонники Linux тоже любят праздники и желают пользоваться всеми благами цивилизации, поэтому проблема не осталась незамеченной. Первым такой режим реализовал Габор Кути [Gabor Kuti] в 1998 году – его заплатка v7c для ядра 2.2.20 до сих пор находится на странице http://falcon.sch.bme.hu/~seasons/linux/swsusp.html, но уже долгое время не поддерживается. Основную работу по реализации спящего режима для Linux провела группа энтузиастов под руководством Найджела Каннингэма [Nigel Cunningham]. Хотя разработка Software Suspend for Linux или swsusp, имеет уже длинную историю, она долгое время рассматривалась как альфа и использовалась энтузиастами на свой страх и риск. Наконец, начиная с версии ядра 2.5.18, код swsusp включен в основное дерево разработки.



Вместе с исходными текстами ядра доступно небольшое описание реализации этого режима: посмотрите файл swsusp.txt в каталоге **/usr/src/linux/Documentation/power**, с которым советую ознакомиться всем для понимания сути проблемы. В настоящее время фактически известны несколько реализаций спящего режима для Linux. Это заплатка *pmdisk* (**CONFIG_PM_DISK**) http://lwn.net/Articles/48471/, сейчас ее код включен в состав *swsusp*, собственно сам *swsusp*, *swsusp2* или *Software Suspend 2* (http://www.suspend2.net/) и *µswsusp* – *User space Suspend* (http://suspend.sourceforge.net/). Первые два проекта в настоящее время интереса не представляют. Зато интересны последние две разработки, развивающиеся сейчас наиболее активно. Проект *µswsusp* относительно молодой (начало разработок датировано апрелем 2006), поддержаны ядра начиная от 2.6.17, реализация – в виде обычной программы, работающей в пространстве пользователя, что дает большие преимущества. Эта разработка уже готова к применению, но некоторые возможности пока не дотягивают до *swsusp2*, о котором и пойдет речь в дальнейшем

В отличие от *µswsusp*, *swsusp2* реализован в виде заплатки к ядру. Сегодня поддержка ядер версии 2.4 этим проектом практически остановлена, последней является заплатка 2.1.5.7B для ядра 2.4.28; работает она стабильно, но дальнейшее развитие не предусмотрено. Аналогичная ситуация и с ядром 2.2. Поддерживаются все файловые системы (некоторые, правда, с оговорками), а также *LVM* и *dm-crypt* (средство для шифрования данных). Работа со спящим режимом про-

Другой подход

Справедливости ради следует отметить, что KDE умеет восстанавливать рабочий стол последнего сеанса, если выйти из него через K > Завершить сеанс. Для настройки такой возможности зайдите в Центр управления KDE > Компоненты > Менеджер сессий и установите галочку напротив Восстанавливать предыдущий сеанс. Правда, приложения все равно закрываются, и рабочий стол восстанавливается лишь частично, хотя этого бывает достаточно, чтобы вспомнить, чем занимался последний раз.

тестирована на многочисленных конфигурациях, хотя возможны и проблемы с некоторым оборудованием. Необходима поддержка расширения процессора PSE или PSE36 (**cat /proc/cpuinfo | grep pse**) – не могу сказать, где ее нет, но с процессорами Intel, AMD и VIA C3 проблем быть не должно. Система сна пока не будет работать с большим количеством оперативной памяти (> 4 ГБ), зато поддерживаются AGP, DRI, USB и прочее. Однако, если жесткий диск не знает ничего о DMA или режиме suspend, что свойственно для ну очень старых дисков, то с данными, скорее всего, придется проститься.

Возможность сохранения данных вместо раздела подкачки в файл, появившаяся во второй версии, имеет особое значение. Представьте себе такую ситуацию: для тестирования запускаете *Knoppix*, который находит на диске swap-раздел, монтирует его и сохраняет в нем свои данные. Если в этом разделе были сохранены результаты suspend, то, естественно, они будут затерты. Аналогичная картина будет наблюдаться при использовании двух ядер, одно из которых не знает о suspend. Использование файла подкачки позволяет избежать таких ситуаций.

Настройка гибернации в Kubuntu

Некоторые производители уже включают возможность перехода в спящий режим в последние версии дистрибутивов. За подробностями конкретно используемого вами дистрибутива отправляю на страницу Wiki проекта SWSup2, где все подробно расписано. В Kubuntu, начиная с версии 6.10, при выходе из системы такой пункт уже имеется. В Kubuntu 6.06 LTS такой возможности по умолчанию нет, но эту несправедливость легко исправить.

К сожалению, если набрать sudo apt-cache search hibernate, можно найти только старую версию скрипта *hibernate*, упрощающего управление этим режимом. Мир не без добрых людей: тем, у кого нет желания самостоятельно компилировать ядро, следует отправиться на страницу http://dagobah.ucc.asn.au/dapper-kernels/, где находится альтернативный репозиторий для Dapper Drake. Добавьте в **/etc/apt/sources.list**:

```
deb http://dagobah.ucc.asn.au/ubuntu-suspend2 dapper/
```

и обновите список пакетов:

```
$ sudo apt-get update
```

Запрос sudo apt-cache search hibernate покажет наличие двух пакетов, которые и нужно установить. Кстати, метапакет *suspend2* включает все необходимое.

```
$ sudo apt-get hibernate suspend2
```

Учитывая «мягкий» характер зависимостей пакетов в Ubuntu, советую внимательно просмотреть, какие пакеты будут установлены дополнительно – и, главное, списки Предлагаемые пакеты и Рекомендуемые пакеты.

Компиляция ядра

В том случае, когда вышеприведенный вариант неприемлем, ядро придется собрать самостоятельно. Исходные тексты ядра можно загрузить с http://www.kernel.org/ или найти на нашем DVD, а с сайта *swasup2* следует взять заплатку под нужную версию.

```
$ cd /usr/src
$ sudo tar -xjvf /home/source/ linux-2.6.22.tar.gz
$ sudo ln -sf linux-2.6.22 linux
$ cd linux
$ wget -c http://www.tuxonice.net/downloads/all/suspend2-2.2.10-for-2.6.22.patch.bz2
$ bzcat suspend2-2.2.10-for-2.6.22.patch.bz2 | patch -p1
```

Теперь в .config активируйте следующие параметры:

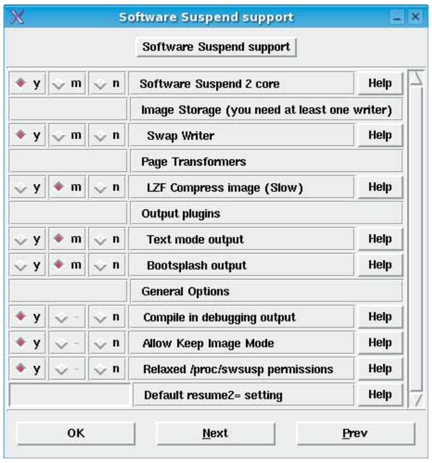
```
CONFIG_SUSPEND2=y
CONFIG_SUSPEND2_FILE=y
CONFIG_SUSPEND2_SWAP=y
CONFIG_CRYPTO_LZF=y
```

Если используется более ранняя версия ядра, желательно, но не обязательно установить и последнюю версию ACPI (http://acpi.sourceforge.net/). См. **Рис. 1**.



Последние приготовления

Также следует позаботиться о том, чтобы раздел подкачки имел достаточный размер (рекомендуется хотя бы двойной размер ОЗУ). Как вариант, можно подготовить файл, в котором и будет сохраняться информация. В конфигурационном файле загрузчика **/boot/grub/menu.lst** к параметрам, передаваемым ядру, добавьте строку **resume2=swap:/dev/hda2**, где **/dev/hda2** – swap-раздел, используемый при гибернации. Теперь, когда система будет засыпать, сигнатура данного раздела будет изменена на S2SUSP, а уже после загрузки и извлечения данных она будет заменена на нормальную – SWASPACE2. И если этот раздел будет использован другим дистрибутивом, то при запуске заснувшей системы будет найдена сигнатура S2SUSP, но нужных данных на разделе не обнаружится, и загрузка будет остановлена. Выходом »





› Рис. 1. Настроить необходимые параметры ядра можно и через графический интерфейс.

» из такой ситуации является загрузка с параметром noresume2 или форматирование раздела заново при помощи *mkswap*, либо использование для хранения данных гибернации файла подкачки вместо раздела.

Загрузите новое ядро. В процессе старта на консоль должны быть выведены сообщения (dmesg | less), вроде:

```
Software Suspend Core.
Software Suspend Swap Writer registered.
Software Suspend 2.2.10: Suspending enabled.
```

Обратите также внимание и на запуск ACPI. Иногда система выводит примерно такое сообщение:

```
ACPI disabled because your bios is from 97 and too old
You can enable it with acpi=force
```

Оно свидетельствует о древности используемого BIOS. Поступите, как подсказывает сообщение, т.е. добавьте параметр acpi=force к параметрам, передаваемым ядру, прописав эти строки в конфигурационном файле загрузчика.

Вперед

Теперь можно укладывать систему спать. Ранее это можно было сделать несколькими вариантами, но в ядрах 2.6, использующих *sysfs*, рекомендуемым является обращение к файлам в **/sys/power/**.

Например, чтобы перевести систему в режим Suspend to RAM, когда питание процессора отключается, но вся информация сохраняется в оперативной памяти, следует ввести

```
$ sudo echo mem > /sys/power/state
```

Для сброса информации в раздел подкачки и последующего отключения – режим Suspend to Disk – используется такая команда:

```
$ sudo echo -n disk > /sys/power/state
```

После загрузки с новым ядром в **/sys/power/** появится подкаталог *suspend2*, файлы в котором позволяют получить информацию о состоянии некоторых параметров системы либо управлять режимом сна. Перевести систему в режим гибернации можно и так:

```
$ sudo echo anything > /sys/power/suspend2/do_suspend
```

после чего компьютер, немного пожужжав, выключится. Если заново запустить систему, то в консоли среди прочих должны появиться сообщения, свидетельствующие о том, что swap-раздел с информацией для восстановления состояния найден.

```
Software Suspend 2.2.10: Swap space signature found.
Software Suspend 2.2.10: This is normal swap space.
```

По умолчанию компрессия и шифрование записываемого образа включены. При необходимости, отключить эти функции можно так:

```
$ sudo  echo 0 > /sys/power/suspend2/compression/enabled
$ sudo  echo 0 > /sys/power/suspend2/encryption/enabled
```

Чтобы опять их включить, достаточно заменить в этих выражениях **0** на **1**.

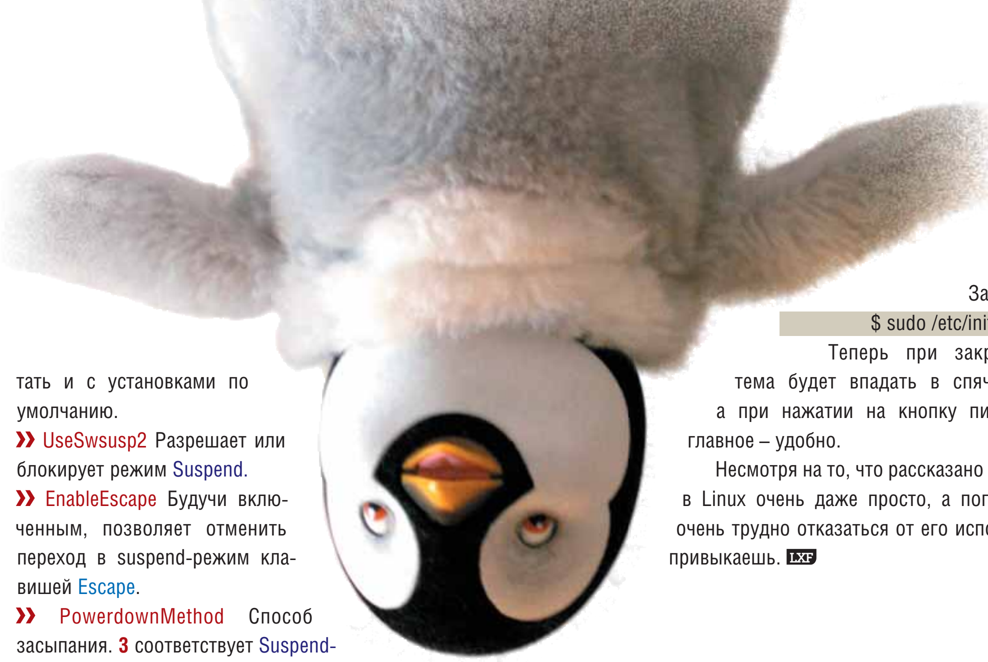
Работа со скриптом hibernate

С помощью файлов, находящихся в **/sys/power/suspend2**, можно задать множество параметров работы системы сна, но вбивать их каждый раз вручную достаточно нудно. Разработчики существенно упростили эту задачу, написав скрипт, делающий весь процесс более наглядным и удобным. Называется он *hibernate* – именно его мы и установили, введя одноименный параметр в apt-get. Просто наберите:

```
$ sudo /usr/sbin/hibernate
```

и система погрузится в режим сна. Скрипт имеет ряд полезных параметров. Например, --save-settings позволит сохранить все настройки в указанный следом файл. Параметр --no-suspend позволит протестировать скрипт без выключения питания. После изменения всех настроек следует протестировать работу, введя в качестве параметра --reboot=1, что заставит систему сохранить все настройки в файл гибернации и перезагрузиться.

Свои параметры скрипт берет из конфигурационного файла **/etc/hibernate/hibernate.conf**, разъяснения по структуре которого можно получить, набрав hibernate с параметром --h. Я приведу здесь лишь наиболее интересные директивы, но в принципе можно рабо-



тать и с установками по умолчанию.

» **UseSwsusp2** Разрешает или блокирует режим Suspend.

» **EnableEscape** Будучи включенным, позволяет отменить переход в suspend-режим клавишей Escape.

» **PowerdownMethod** Способ засыпания. **3** соответствует Suspend-To-RAM, **4** – гибернации, **5** – отключению питания. Для ноутбуков разумно использовать последний вариант.

» **IncompatibleDevices** и **IncompatiblePrograms** Первое – это список устройств, несовместимых с гибернацией (обычно **/dev/dsp, /dev/video***). Процессы, имеющие к ним доступ, будут перед засыпанием остановлены. Второе – названия несовместимых программ; они будут остановлены независимо от того, к каким устройствам обращаются.

» **Unmount** и **UnmountFSTypes** Перечень файловых систем, которые следует размонтировать перед переходом в suspend-режим. Хорошей идеей будет включить в него используемые вами сетевые ФС. Первая директива позволяет указать точки монтирования, а вторая – типы файловых систем (например, smbfs).

» **StartServices**, **StopServices**, **RestartServices** Список сервисов, которые необходимо остановить при выключении и запустить/перезапустить при включении.

Кстати, в некоторых дистрибутивах (Kubuntu как раз относится к ним) для удобства все настройки разнесены по нескольким файлам, которые подключены в **hibernate.conf** директивой Include.

Теперь можно пробовать залечь в спячку. Для начала рекомендуется сделать это в консоли. Введите init 3 и запустите скрипт.

```
$ sudo /usr/ sbin/hibernate
  Good night!
  Good morning!
```

Если все прошло удачно, попробуйте запустить его из-под X. Если и это получилось, можете себя поздравить.

Немного автоматизируем процесс

На этом статью о гибернации в Linux можно бы и закончить, но еще пару слов хочется добавить для удобства использования. Естественно, каждый раз запускать скрипт не всегда удобно, иногда бывает проще закрыть крышку ноутбука, а вновь включив питание, обнаружить все на своих местах. Для этого необходимо использовать демон *acpid*, который является пользовательским интерфейсом, позволяющим управлять любыми событиями ACPI, доступными через **/proc/acpi/event**. При этом *acpid* читает набор конфигурационных файлов в каталоге **/etc/acpi/events/**.

Скорее всего, необходимый пакет уже есть в вашем дистрибутиве, и надо просто проверить его наличие и создать конфигурационные файлы. Если же пакета нет, скачайте последнюю версию с сайта http://phobos.fs.tum.de/acpi/, скомпилируйте и установите. В каталоге **/etc/acpi/events** создайте два файла: **lid** и **power**. Первый описывает реакцию на закрытие крышки, второй – на нажатие кнопки включения питания.

В **/etc/acpi/events/lid** впишите

```
event=button/lid.*
action=/usr/sbin/hibernate
```

Аналогично поступите и с **/etc/acpi/events/power**

```
event=button/power.*
action=/sbin/shutdown -h now
```

Затем перезапустите *acpid*.

```
$ sudo /etc/init.d/acpid restart
```

Теперь при закрытии крышки ноутбука система будет впадать в спячку с выключенным питанием, а при нажатии на кнопку питания выключаться. Просто, а главное – удобно.

Несмотря на то, что рассказано много, настроить спящий режим в Linux очень даже просто, а попробовав один раз на практике, очень трудно отказаться от его использования. К хорошему быстро привыкаешь. LXF





Переходим

ЧАСТЬ 3 В прошлый раз мы рассмотрели особенности печати книги на ризографе. Настало время добиться наивысшего качества. В этом нам поможет офсетная печать и **Александр Чернышов**.

Вопреки распространённому убеждению, коего я сам до недавнего времени придерживался, что офсет – это дорого и только для избранных, на деле всё оказалось значительно интереснее. Если вам надо качественно напечатать книгу тиражом более 300 экземпляров, то имеет смысл поискать офсетную типографию и попросить, чтобы вам посчитали стоимость тиража вашей книги. Обычно услуга эта бесплатная.

Многие типографии «дерут» высокие цены. Но при желании и везении вполне можно найти недорогую офсетную типографию. Так мне однажды повезло с типографией в Москве: книга в качественном жёстком переплёте, тиражом 500 экземпляров, в которой было почти 800 страниц (без цветных иллюстраций), обошлась меньше чем по 300 рублей за экземпляр, что я считаю очень неплохим показателем.

Что же касается качества, то офсет – это как раз тот случай, когда все положительные свойства TeX'а проявляются оптимальным образом. В конце концов, именно для подобных технологий печати тиражей TeX и создавался. Так что – осваиваем офсет.

Формат бумаги

Вы, вероятно, будете удивлены применением в типографиях собственных стандартов размеров бумаги. Вместо «A4» и «A5», вам назовут что-то вроде «60x90/8» и «60x90/16». Кстати, «промежуточный» формат большинства книг по программированию имеет здесь своё стандартное обозначение: «70x100/16» (читается «семьдесят на сто в шестнадцатую долю»).

Дело в том, что офсетная машина печатает сразу целый печатный лист. Это большой лист бумаги, на котором напечатаны все страницы одной книжной тетради. Причём лист, запечатанный только с одной стороны, называется оттиском, а с обеих сторон – собственно печатным листом. Первые две цифры (например, 70x100) описывают размер печатного листа в сантиметрах.

Далее лист фальцуют (складывают) в несколько раз, получая сразу тетрадку книги. Её теперь остаётся только прошить и обрезать. Параметр доли как раз и определяет, сколько страниц текста будет напечатано на одном печатном листе. Так, /16 означает, что на оттиске будет 16 страниц, а на печатном листе (и в одной тетрадке книги) – 32 страницы.

Между прочим, именно в силу этого вы должны обеспечить, чтобы количество страниц в вашей книге было кратным 16 (я не ошибся – по технологическим причинам именно 16). По счастью, в TeX'е можно слегка «поиграть» параметрами междустрочного интервала (\baselineskip, \lineskiplimit, \lineskip) и добиться желаемого. Лишь в редких случаях приходится поступать «по старинке» и выбрасывать «лишний» материал либо дописывать «недостающий».

При подготовке вёрстки в TeX'е важно правильно задать формат гранки. Исходите из того, что после фальцовки на обрезку с каждой стороны уйдёт примерно по 1–1,5 см. Прикиньте, какие вам бы хотелось оставить в книге поля, где у вас будет располагаться колонцифра. Можно просто замерить линейкой гранку в готовой книге подходящего формата. В конце концов вы получите два числа, которые надо задать как \hsize и \vsize.

Спуск

Чтобы после фальцовки все страницы в тетрадке следовали в нужном порядке, их надо правильно расположить на каждом печатном листе. Процесс этого расположения называется спуском. В принципе, в современном Linux есть средства, позволяющие выполнить спуск самостоятельно (мы рассмотрим эту возможность позже), но обычно в этом нет необходимости. В большинстве случаев типография делает спуск сама.

Печать плёнок

А вот теперь переходим к самому интересному. Любой офсет начинается с плёнок. Именно «в плёнках» затем делают спуск и дальнейший процесс от нас уже не зависит. Как только плёнки взяты типографией в работу, можете мыть руки, чтобы не испачкать ими сигнальные экземпляры.

Однако, качество результата в наибольшей степени зависит именно от качества вывода плёнок. А здесь от нас зависит очень многое.

В зависимости от типографии вам могут предложить:

- принести свои плёнки (и вам придётся печатать их самостоятельно);
- изготовить плёнки по вашему файлу на оборудовании типографии.

Рассмотрим каждый из этих вариантов.

...В типографии

«...и тут пошли огрехи всей системы...»
Из монолога Жванецкого

Вообще, этот вариант на мой взгляд предпочтительнее. Типография, предлагая его, обычно «имеет за душой» довольно мощное печатающее устройство (2400, или даже 4800 dpi). В результате итоговое качество будет соответствовать непревзойдённому металлическому набору. И к этому надо стремиться. Тем более что в арсенале нашего TeX'а имеется возможность делать шрифты разрешением аж до 9600 dpi! Только узнайте о возможностях устройства заранее и постарайтесь подготовить свой электронный оригинал-макет с соответствующим разрешением. Например:

Месяц назад Мы подробно рассмотрели печать на ризографе.

на офсет

```
dvips -D 2400 ..... -o file.ps file.dvi
```

При работе программы *dvips* обращайте внимание на все предупреждения о генерации шрифтов! Иногда встречаются шрифты со скрытыми ошибками, а также для некоторых букв в некоторых шрифтах случаются банальные переполнения программы *metafont*. Это приводит к потере букв и даже целых участков текста, набранных забракованным шрифтом, или к непредсказуемым заменам. Бороться с этим можно только понижением разрешения или переходом на векторные шрифты (скажем, CM-Super – мы обсудим их чуть позже).

Ключ -O не нужен – окончательное расположение материала в спуске всё равно делается вручную. Главное, чтобы печатаемый материал в принципе попадал в зону печати. И ключ -x тоже не нужен. Для офсетной печати вывод всегда делается в масштабе 1:1.

Казалось бы, после получения **file.ps** все проблемы должны быть сняты. Но оказывается, что это не всегда так – подробности читайте во врезке.

...И самостоятельно

Этот вариант может показаться дешевле, но для получения высокого качества потребует от вас учёта дополнительных факторов, которые мы сейчас и обсудим.

Сначала о принтерах. Годятся только лазерные. Чем выше разрешение принтера, тем лучше. Настоящего профессионального качества в «домашних» условиях, конечно, не добиться, но если вы найдёте лазерный принтер с разрешением 1200 dpi, понимающий PostScript и с новым (не перезаправленным!) картриджем, да ещё с барабаном в пределах заводского ресурса, то вам крупно повезло – вы сможете получить плёнки почти профессионального качества.

Если принтер не понимает PostScript, то придётся использовать GhostScript (в Linux), либо переводить оригинал-макет в PDF и печатать из Adobe Reader (в Windows). Это способно очень сильно увеличить время вывода каждой страницы при высоком разрешении печати. Для ускорения можно попытаться использовать векторные шрифты вместо стандартных, но это далеко не всегда возможно.

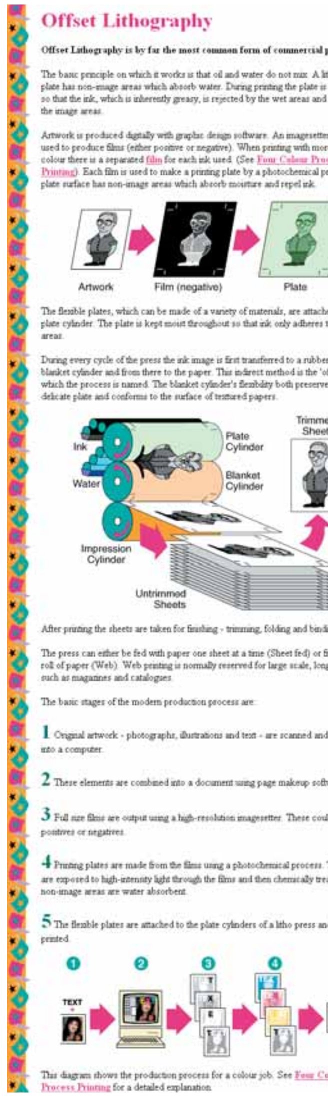
Помните также, что желательно настроить *metafont* на растеризацию шрифтов именно для вашей модели принтера!

Векторные шрифты

Технология, используемая в шрифтах *TeX*, иногда приводит к неприятностям. Итоговые шрифты остаются растровыми, и при попытке их генерации для больших разрешений *metafont* иногда переполняется (не хватает длины его разрядной сетки). То есть, используя технологию *metafont*, эти шрифты получить в высоком полиграфическом качестве невозможно.

Поэтому сообщество пользователей *TeX* постоянно предпринимает усилия в направлении использования в *TeX* векторных аналогов применяемых шрифтов. Отрадно, что в последнее время уделили, наконец, внимание и кириллическим шрифтам.

На данный момент разработан и постепенно проникает «в массы» пакет шрифтов под названием *CM-Super*. В стандартные дистрибутивы



Своенравный ps2pdf

Мне надо было напечатать на офсете книгу, вёрстка которой была рассчитана на формат 70x100/16 (размер страницы не кратен А4). Типография бралась сама изготовить плёнки по моему **file.ps**. Но вдруг обнаружилось, что их специализированная программа управления принтером плёнок упорно считает, что каждая страница моей книги расположена на листе А4. Это приводило к значительному перерасходу дорогостоящей плёнки.

Переплачивать за плёнку не хотелось, и я решил поискать решение. Первое, что пришло на ум – ключ -T команды dvips, задающий (вроде бы) размер листа. Но этот способ проблемы не решил.

Второе решение – «правильно» установить параметр BoundingBox – оказалось вообще несостоятельным. Дело в том, что этот «параметр» на самом деле записывается в комментариях файла PostScript и используется только программами типа *TeX*, которые сами PostScript не интерпретируют. Профессиональные же программы, интерпретирующие PostScript, комментарии игнорируют.

Проблема оказалась в том, что размер листа вообще не попадал в мою вёрстку, и по умолчанию интерпретатор PostScript принимал его равным А4. Решение проблемы следующее:

1 Размер листа надо задавать внутри верстаемого документа (в его начале) командой

```
\special{papersize=WIDTH,HEIGHT}
```

где вместо WIDTH и HEIGHT подставить необходимые значения.

В моём случае команда выглядела так

```
\ special{papersize=164mm,235mm}
```

После этого получить DVI-файл обычным способом.

2 Найти в системе файл конфигурации драйвера **dvips config.ps**. Найти в нём определение формата печати

```
@ unknown 0in 0in
```

Если это определение есть, то вам повезло. Если же его там нет, создайте в домашнем подкаталоге файл **.dvipsrc** со следующим содержимым:

```
@ unknown 0in 0in @+ statusdict /setpageparams known { hsize vsize 0 1 statusdict begin { @+ setpageparams } stopped end } { true } ifelse { statusdict /setpage known @+ { hsize vsize 1 statusdict begin { setpage } stopped pop end } if } if
```

3 Теперь можно переводить вёрстку в PostScript:

```
dvips -D 2400 -t unknown -O 0.5cm,0cm -o file.ps file.dvi
```

Параметр -O в данном случае нужен именно для того, чтобы попасть в зону печати.

Попутно выяснилось, что проверить правильность выполненной работы довольно легко, «не выходя» из Linux. Достаточно скомандовать:

```
ps2pdf file.ps file.pdf
```

и далее

```
gv file.pdf
```

В этом случае просмотр осуществляется по действительным размерам страницы, заданным непосредственно в PS-программе.

Linux он, кажется, пока не входит, но его можно взять из сети Интернет (**CTAN:fonts/ps-type1/cm-super** или **ftp://ftp.vsu.ru/pub/tex/font-packs/cm-super**) и попытаться поставить в своей системе.

Инструкция по установке, идущая в комплекте с пакетом, не совсем верная. Но с помощью Ивана Чернявского мне удалось заставить работать этот пакет на своей системе.

Пакет включает полный комплект векторных шрифтов, позволяющий заменить все основные «стандартные» шрифты *TeX*. Причём шрифты идентичны со своими аналогами по метрике, так что при их замене вёрстка не «уплывает». После его установки для пользователей *LaTeX* не меняется (теоретически) вообще ничего (если только в вёрстке не используются «нестандартные» для *LaTeX* шрифты). Пользователям же plain *TeX* придётся привыкать к новым именам шрифтов – так, вместо lhr10 придётся теперь писать larm1000. А заодно придётся «передвинуть» по кодам некоторые русские знаки, например «номер».

Вообще, пакет оставил приятное впечатление. Получающийся PostScript просто идеален: хорошо конвертируется в PDF, просматривается Acrobat и печатается. При этом для нас очень важно, что теперь он вообще не зависит от разрешения принтера! Правда, и жирностью мы теперь управлять не можем: растеризация этих шрифтов – дело самого принтера.

Однако все эти прелести работают, только если в документе не используются нестандартные шрифты. Например, если вы верстаете книгу по шахматным партиям и используете макропакет chess, то шрифты этого макропакета всё равно будут растровыми.

Если же ваши нестандартные шрифты ещё и русские буквы содержат (а вы эти буквы используете), то у вас могут возникнуть проблемы и похуже – дело в том, что внутренние кодировки шрифтов семейства lh (стандартных для большинства русификаций *teTeX*), а также других шрифтов на их базе и шрифтов CM-Super различаются. В результате совмещать эти шрифты в одной вёрстке нельзя.

Снова о плёнках

Теперь осталось обсудить собственно процесс вывода.

В зависимости от принятой в типографии технологии, вас попросят изготовить либо прозрачные плёнки, либо полупрозрачные (иногда их называют кальками). Технология печати в обоих случаях немного различается, но сэкономить всё равно не удастся, так как, независимо от формата вашей книги, печатать вы будете в масштабе 1:1 на формате A4. Посему постарайтесь центрировать свои страницы на плёнке. Ширина остающихся белых полей играет очень большую роль. Без полей плёнки невозможно будет закрепить в спуске. А лишние поля в типографии просто обрежут.

» **Прозрачные плёнки**. Здесь всё просто. Печать выполняется точно так же, как на бумаге. Весь процесс уже описан раньше, и больше рассказывать нечего.

» **Полупрозрачные плёнки**. Здесь вывод надо выполнить в зеркальном отражении. Для этого надо добавить в итоговый документ PostScript программу зеркального разворота.

В состав современных установок *teTeX* входит файл **mirr.hd**, содержащий эту программу. Применяется он так (искать его не надо – если он есть, программа *dvips* должна найти его сама):

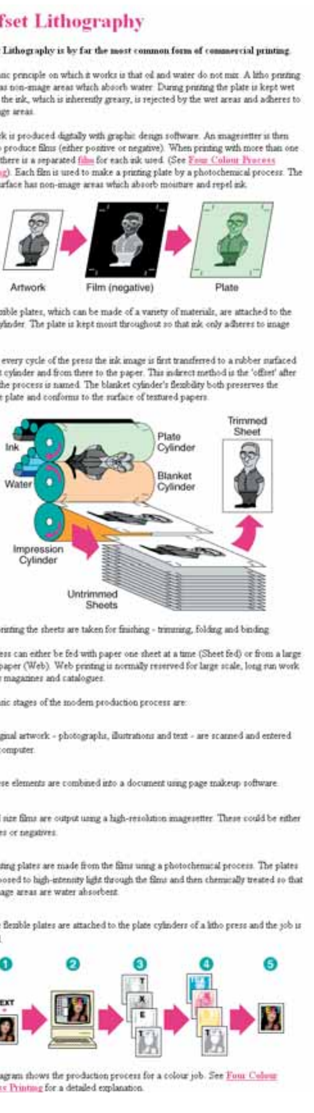
```
dvips -hmirr.hd [другие ключи] -o file.ps file.dvi
```

Вот и всё!

Делаем спуск сами

Повторю ещё раз: вам вряд ли это понадобится. Но технология может пригодиться для других целей [например – верстки «книжкой» и печати на обычном лазерном принтере методички, автореферата или даже одного из многочисленных руководств texinfo, – *прим. ред.*].

Вся работа выполняется командой pstops, обычно входящей в пакет *psutils*, существующий практически в любом дистрибутиве Linux. Вообще-то её основное предназначение – пересортировка страниц в готовом документе PostScript, но и разложить несколько страниц на большом листе она тоже может.



Ключей командной строки у команды не так много, и они достаточно хорошо описаны в man.

Самая сложная часть команды – описание раскладки страниц (pagespec).

Вот как она представлена в документации:

```
pagespec = [modulo:]specs specs = spec[+specs][,specs] spec = [-]pageno[L][R][U][@scale][(xoff,yoff)]
```

В этом описании:

» modulo (модуль) – количество страниц на одном листе (блоке), включая те, которые на данном проходе просто пропускаются, например, для печати на обратной стороне.

» spec – описание вывода и расположения на листе одной страницы с номером pageno (по модулю modulo, то есть pageno должен иметь значение от **0** до **modulo–1**).

» xoff и yoff задают смещение страницы на листе, допускается указание величины смещения в сантиметрах (cm), дюймах (in) или специальных единицах w и h, соответствующих ширине и высоте страницы (их размеры должны быть заданы ключами командной строки).

» **Параметры** L, R и U задают поворот страницы влево, вправо и вверх ногами; точкой поворота считается левый нижний угол страницы.

» scale задаёт масштабирование страницы и для наших целей не нужен.

» **Знак** минус задаёт реверсивный порядок вывода страниц.

Параметры, объединённые знаком +, описывают расположение страниц на одном листе, объединённые через запятую – на разных (это может пригодиться, если у вас двухсторонний принтер).

Ключевым к нашей цели будет следующий пример:

```
4:2U(1w,2h)+0(0w,0h)
```

В нём мы взяли первый и третий из каждых четырёх листов документа и разместили третий лист над первым, причём перевернули его вверх ногами. Теперь переходим к собственно спуску.

Вообще говоря, порядок страниц в спуске зависит от многих факторов. Поэтому приводимый ниже пример носит демонстрационный характер. Реальное расположение страниц в спуске вы всегда должны определять на месте.

Выполним спуск в шестнадцатую долю (/16). При этом в одной тетради будет 32 страницы. Расположим страницы на первом оттиске так:

```
17 16 9 24 32 1 8 25 29 4 5 28 20 13 12 21
```

а на втором оттиске (с учётом печати на противоположной стороне!) так:

```
23 10 15 18 26 7 2 31 27 6 3 30 22 11 14 19
```

В обеих раскладках в первой и третьей строках страницы должны быть перевёрнуты вверх ногами.

Чтобы выполнить раскладку, подготовим два вспомогательных файла.

Первый назовём **pages.first** (весь файл – одна строка!):

```
32:16U(1w,4h)+15U(2w,4h)+8U(3w,4h)+23U(4w,4h)+31(0w,2h)+0(1w,2h)+7(2w,2h)+24(3w,2h)+28U(1w,2h)+3U(2w,2h)+4U(3w,2h)+27U(4w,2h)+19(0w,0h)+12(1w,0h)+11(2w,0h)+20(3w,0h)
```

второй – **pages.second** (опять – одна строка!):

```
32:22U(1w,4h)+9U(2w,4h)+14U(3w,4h)+17U(4w,4h)+25(0w,2h)+6(1w,2h)+1(2w,2h)+30(3w,2h)+26U(1w,2h)+5U(2w,2h)+2U(3w,2h)+29U(4w,2h)+21(0w,0h)+10(1w,0h)+13(2w,0h)+18(3w,0h)
```

Теперь, чтобы выполнить спуски для всех первых оттисков, достаточно дать команду

```
pstops -h... -w... `cat pages.first` file.ps first.ps
```

и в файле **first.ps** будут находиться последовательные листы первых оттисков всех печатных листов. Обратите внимание, что необходимо задать правильные значения -h и -w – они должны быть равны размеру одной страницы с полями, причём с запасом на фальцовку и обрезку.

Вторые оттиски могут быть получены командой

```
pstops -h... -w... `cat pages.second` file.ps second.ps
```

Просмотреть результат можно командой gv, например,

```
gv first.ps
```

при этом весь лист можно увидеть только на формате A0, который в конфигурации программы по умолчанию закрыт. Разрешите его, отредактировав файл **/etc/X11/app-defaults/GV** (с правами root), а затем выберите в меню форматов запущенной программы.

Приведённый пример имеет целый ряд недостатков. Во-первых, в правильном спуске поля между страницами у корешка и у обреза разные, а у нас они одинаковые, так что для реального применения промежуточные файлы с раскладкой придётся доработать. Во-вторых, не согласован порядок следования оттисков в файлах **first.ps** и **second.ps**. Но для печати плёнок это не имеет значения: формы всё равно получают раздельно. В-третьих, если количество страниц в книге кратно 16, но не кратно 32, одна тетрадь должна будет иметь половинную толщину (обычно это третья от конца книги тетрадь). Наш пример этого не учитывает. Но, опять же, при желании его можно доработать, написав дополнительные промежуточные файлы раскладок.

Наконец, самое важное. При перекладке страниц программа немного переписывает исходный код PostScript в документе, в результате чего заданный в документе истинный размер страницы (который мы выше вставили с таким трудом) будет завуалирован и не сможет правильно распознаться. При этом размер листа, на котором выполняется спуск, в документ вставлен не будет. А для профессиональных программ печати это уже проблема.

Ее можно решить исправлением итогового PS-файла вручную. Я приведу упрощённый пример, имеющий под собой реальную основу. Итак, плёнки моей книги формата 70x100/16 выводили на машине, имевшей ширину печати сразу в три моих страницы. Переложить страницы для этого случая было довольно просто (реально использованная команда):

```
pstops -h 235mm -w 164mm '3:0(0w,0h)+1(1w,0h)+2(2w,0h)' infile.ps ofile.ps
```

Далее файл **ofile.ps** был открыт обычным текстовым редактором и в нём были сделаны следующие изменения:

1 увеличена втрое ширина (точнее, правая граница листа) в %%BoundingBox (конкретно, с 465 до 1395);

2 символами % закомментирована строка с /setpagedevice и следующая за ней (именно она снова заставляет интерпретатор PostScript работать с листом A4, а у нас свой размер страницы);

3 в строке (найти поиском) TeXDict begin 30580752 43819980 1000 600 600 .... первое число (как раз ширина листа) заменено втрое большим (91742256).

Теперь преобразованный документ готов к печати на плёнке.

Предварительную проверку результата можно выполнить, как и ранее:

```
ps2pdf ofile.ps ofile.pdf
gv ofile.pdf
```

Если на одном широком листе видны все три страницы в ряд, значит, всё сделано правильно.

Вот, пожалуй, и всё, что мне хотелось рассказать сегодня.

Happy TeX'ing! LXF

Linux в сети WPA:

ЧАСТЬ 2 Теперь, когда в вашем распоряжении есть защищенная по максимуму Wi-Fi-сеть, дело за малым: осталось только подключить к ней свой собственный ноутбук. Этим и займется **Андрей Боровский**.

В прошлой статье мы познакомились с сервером FreeRADIUS и дополнительными возможностями авторизации WPA Enterprise, которые он предоставляет. На этот раз речь пойдет о настройке авторизации беспроводных клиентов Linux в сетях, использующих авторизацию WPA Enterprise. Конечно, в вашей беспроводной сети наверняка есть и Windows-машины, но останавливаться на методах настройки Windows мы не будем. Настройка авторизации клиентов Windows в беспроводной сети (в том числе и методами WPA) довольно проста и подробно документирована (пошаговые руководства на русском языке можно найти и в Сети). В мире Linux дела обстоят несколько сложнее. Впрочем, как мы увидим, не намного.

Самая серьезная проблема, с которой вы столкнетесь при подключении Linux-машины к беспроводной сети, заключается в подборе драйвера, поддерживающего вашу Wi-Fi-карту. Ситуация с поддержкой чипсетов беспроводных карт в ОС Linux сейчас напоминает положение дел с аппаратным ускорением графики в 2000–2001 годах: необходимые программные компоненты уже есть, но поддержка конкретного оборудования реализована не полностью. Справедливости ради надо отметить, что ситуация улучшается довольно быстро (см. врезку).

При описании настройки авторизации беспроводного Linux-клиента я буду исходить из предположения, что вы уже прошли мучительный процесс поиска и установки Linux-драйвера для вашей Wi-Fi-карты (или подобрали поддерживаемую Linux Wi-Fi-карту) и научились «поднимать» беспроводной сетевой интерфейс в своей системе.

Аутентификацией WPA в Linux управляет демон *wpa_supplicant*. Перед началом настройки беспроводного клиента WPA-enterprise убедитесь в том, что этот демон установлен в вашей системе. Помимо него, нам понадобятся демон управления сетями *NetworkManager*, утилита *wavemon*, а также, в зависимости от того, по какому пути пойдет настройка – графическая оболочка GNOME NetworkManager Applet [или ее KDE-аналог – K*NetworkManager*, – прим. ред.], утилиты *wpa_gui* и *KWlan*. Все перечисленные программы можно найти в любом современном дистрибутиве Linux. Многие системы устанавливают их по умолчанию, но в некоторых случаях нужные пакеты придется добавлять самостоятельно.

Выбираем беспроводной адаптер

Изначально распространение открытых драйверов беспроводных адаптеров для Linux сдерживалось теми же причинами, что и распространение открытых драйверов графических карт с аппаратной поддержкой OpenGL: производители соответствующего оборудования опасались утечки своих фирменных секретов. Компания Atheros Communications, производитель весьма распространенных Wi-Fi-чипсетов, предлагает разработчикам драйверов для Linux и BSD бесплатный, но закрытый компонент, создающий единый программный интерфейс для различных чипсетов Atheros (Hardware Abstraction Layer, HAL). В то же время в мире открытых ОС ведется разработка полностью открытого компонента абстракции оборудования (OpenHAL). Atheros OpenHAL изначально создавался для OpenBSD, однако в настоящее время активно портируется на платформу Linux. Правовой аспект разработки открытого компонента, тесно интегрированного с фирменными разработками Atheros, пока что не совсем ясен, но ситуация обнадеживает. Более подробные сведения о разработке OpenHAL можно получить по ссылке **madwifi.org/wiki/OpenHAL**.

В настоящее время наиболее полная поддержка Linux обеспечена чипсетам Intel, Atheros и Ralink Technology. Не следует также забывать о постепенно выходящем из употребления драйвере *ndiswrapper*, который является Linux-оберткой для Wi-Fi-драйверов, написанных для Windows.

Настройка WPA с помощью NetworkManager

Начнем с самого простого и приятного способа настройки аутентификации в беспроводной сети. Современные дистрибутивы Linux содержат компонент *NetworkManager*; его главная задача заключается в том, чтобы обеспечить работу сети прямо из коробки. *NetworkManager* – это демон, который выполняет роль связующего звена между такими компонентами, как демон аутентификации *wpa_supplicant*, клиент DHCP и другими. Во многих системах *NetworkManager* будет запущен сразу после установки – проверить это можно с помощью утилиты *nm-tool*, о которой будет сказано ниже. Если *NetworkManager* не запускается автоматически, его следует запустить в режиме root командой *NetworkManager*. Если вы хотите получить консольный вывод *NetworkManager* (например, в случае возникновения ошибок), запускайте программу с ключом **--no-daemon**. Учтите, что файлы конфигурации, оставшиеся от предыдущих попыток настроить подключение к беспроводной сети другими способами, могут помешать *NetworkManager* определить параметры оборудования и сети. Например, мне, чтобы запустить *NetworkManager*, пришлось удалить файл **/etc/network/interfaces** со старыми настройками. После запуска демона вы можете проверить его работу с помощью утилиты *nm-tool*. Если с *NetworkManager* все в порядке и утилита *nm-tool* распечатала на экране консоли корректные сведения о ваших сетевых интерфейсах и сетях, к которым они подключены, можете плясать и петь – основная часть работы по настройке подключения к сети вами уже сделана.

Для управления *NetworkManager* существует несколько графических утилит, наиболее функциональной из которых является апплет *NetworkManager* для GNOME. Утилита настройки *NetworkManager* от GNOME запускается командой *nm-applet*. Если ее нет в вашей системе, значит, скорее всего, соответствующий пакет просто не установлен. Если апплет *NetworkManager* отказывается запускаться с сообщением "Failed to open device XXX", вам следует перезапустить демон *dbus*:

```
sudo /etc/init.d/dbus restart
```

Если же апплет *NetworkManager* запустился успешно и обнаружил настройки беспроводной сети, в системном лотке рабочего стола появляется пиктограмма с несколькими точками и бегущими по кругу стрелками. Щелчок мышью по этой пиктограмме открывает окно, в котором перечислены обнаруженные сети. Справа от имени каждой беспроводной сети расположен значок, указывающий на наличие

Месяц назад Мы научились настраивать сервер FreeRADIUS.

есть контакт!

защиты сети, и индикатор качества сигнала. Чтобы подключиться к одной из обнаруженных сетей, следует щелкнуть по ее имени. Правда, по умолчанию для сетей WPA апплет *NetworkManager* предлагает только аутентификацию WPA Personal, то есть WPA-PSK. Для подключения к сети с использованием других методов аутентификации выберите команду Connect to Other Wireless Network.... В открывшемся окне (**Рис. 1**) можно ввести необходимые параметры. Для настройки аутентификации по протоколу PEAP MS CHAP V2 вам понадобятся имя пользователя сети и пароль, а для настройки авторизации методом EAP TLS – файлы сертификатов, ключа и пароль к секретному ключу. Пояснения смысла полей ввода этого окна вы найдете ниже, в разделе «Настройка подключения к сети».



› **Рис. 1. Подключение к сети WPA Enterprise с помощью апплета *NetworkManager*.**

Убедившись, что подключение к сети с помощью *NetworkManager* работает, вы можете добавить *NetworkManager* в rc-скрипты для его автоматического запуска в процессе загрузки ОС.

Недостатком апплета *NetworkManager* на сегодняшний день является то, что утилита не умеет настраивать параметры IP-адреса сетевого интерфейса. Отсутствие средств такой настройки не является проблемой, если вы используете DHCP во всех сетях, к которым подключаетесь, но многие, в том числе и я, используют в домашней сети статические адреса. Настроить статический IP для соединения, управляемого *NetworkManager*, можно стандартными средствами вашего дистрибутива Linux, однако тут возникает еще одна проблема – хотя штатные средства настройки сетевых интерфейсов ОС обычно позволяют создавать несколько профилей (DHCP для одной сети, статический адрес – для другой), эти профили нельзя привязать к профилям *NetworkManager*, так что при переключении с одной сети на другую (которое *NetworkManager* выполняет автоматически) профили настроек сетевых интерфейсов придется менять вручную. Разработчики *NetworkManager* обещают добавить поддержку настроек IP-адресов в следующую версию (0.7) своей утилиты, а пока что придется мириться с этим неудобством.

Настройка wpa_suppplicant

Если простая настройка сети с помощью *NetworkManager* у вас не работает, придется прибегнуть к ручной настройке *wpa_supplicant*. В системе Linux *wpa_supplicant* (который, кстати, доступен и на платформе Windows) обычно запускается как демон, поэтому мы будем называть его «демон аутентификации». Демон аутентификации *wpa_supplicant* отвечает за решение двух задач безопасного беспроводного соединения – безопасной аутентификации по протоколам WPA/WPA2 и управления ключами, используемыми при шифровании передаваемых данных.

Поскольку *wpa_supplicant* и сопутствующие ему модули развиваются очень быстро, всегда есть вероятность, что оборудование, не поддерживаемое старой версией *wpa_supplicant* или Wi-Fi-драйвера, будет поддерживаться более новыми версиями. Если экземпляр *wpa_supplicant* из вашего дистрибутива не сумел наладить взаимодействие с вашей беспроводной картой, попробуйте собрать программу из исходных текстов последнего релиза (который всегда доступен на официальном сайте **hostap.epitest.fi/wpa_supplicant/**). Для успешной сборки *wpa_supplicant* вам понадобятся исходные тексты драйверов (обновление драйвера может решить проблему взаимодействия *wpa_supplicant* и беспроводной карты). Исходные тексты драйверов следует загружать отдельно, с сайтов разработчиков. Список поддерживаемых драйверов, как и ссылки на сайты, можно найти на той же официальной странице.

Демон *wpa_supplicant* имеет модульную структуру, так что если вы собираете его из исходных текстов, вы должны убедиться, что в процессе сборки были подключены все необходимые модули. Список модулей, подключаемых к *wpa_supplicant* в процессе сборки, должен содержаться в файле **.config**, расположенном в корневой директории исходных текстов *wpa_spplicant* (по умолчанию, в корневой директории расположен файл **defconfig**, из которого перед началом сборки необходимо сделать файл конфигурации, подходящий для вашей системы). Файл **.config** состоит из набора строк вида:

```
CONFIG_<опция>=y
```

Убедитесь, что ваш файл **.config** включает строки

```
CONFIG_EAP_PEAP=y
CONFIG_EAP_TTLS=y
```

Они автоматически подключают модули, необходимые для поддержки аутентификации с помощью MS-CHAP V2 и цифровых сертификатов (EAP-TLS). Кроме списка модулей, реализующих методы и протоколы аутентификации, файл **.config** должен включать список поддерживаемых интерфейсов драйверов – например,

```
CONFIG_DRIVER_MADWIFI=y
CONFIG_DRIVER_NDISWRAPPER=y
CONFIG_DRIVER_WEXT=y
```

Проследите за тем, чтобы демон аутентификации был сконфигурирован с поддержкой интерфейса драйвера вашей Wi-Fi-карты. Двоичный пакет *wpa_supplicant*, входящий в ваш дистрибутив Linux, сконфигурирован с поддержкой всех доступных демону Wi-Fi-драйверов. Это сделано с целью обеспечить максимальную совместимость

двоичного пакета, однако в некоторых случаях драйверы конфликтуют между собой. Если вы никак не можете запустить процесс аутентификации и все другие средства не помогают, попробуйте удалить из файла конфигурации все директивы подключения драйверов, кроме одной, фактически используемой вашей беспроводной картой.

Поскольку мы настраиваем *wpa_supplicant* вручную, нам придется самостоятельно «привязать» наши настройки к сетевому интерфейсу – например, в Kubuntu в файл **/etc/network/interfaces** следует добавить запись вида:

```
auto ath0
iface ath0 inet static
address 192.168.1.36
netmask 255.255.255.0
gateway 192.168.1.1
wpa-driver madwifi
wpa-conf /etc/wpa_supplicant.conf
```

Здесь ath0 – имя настраиваемого беспроводного сетевого интерфейса. Далее следуют статические адреса узла и шлюза. Если узлы вашей сети получает адреса по DHCP, вместо этих настроек надо написать

```
iface ath0 inet dhcp
```

Последние две строчки настраивают собственно параметры WPA. В атрибуте wpa-driver указывается имя драйвера беспроводного адаптера, а в атрибуте wpa-conf – путь к файлу конфигурации *wpa_supplicant*. В некоторых дистрибутивах Linux (и в старых версиях Kubuntu) параметры запуска *wpa_supplicant* следует прописывать в специальном файле **/etc/sysconfig/wpa_supplicant** (в старых версиях Kubuntu – **/etc/default/wpasupplicant**). Вот типичный пример такого файла:

```
INTERFACES="-iwlan0"
DRIVERS="-Dmadwifi"
```

Значением атрибута INTERFACES должен быть список интерфейсов устройств, для которых запускается демон *wpa_supplicant*, а значением атрибута DRIVERS – список интерфейсов необходимых драйверов.

Настройка подключения к сети: PEAP MSCHAP V2 и EAP-TLS

Linux предоставляет много способов настройки аутентификации с помощью *wpa_supplicant*, в том числе с применением вспомогательных утилит. Мы начнем с настройки путем непосредственного редактирования файлов конфигурации. Вспомогательные утилиты настройки авторизации (включая апплет GNOME NetworkManager) используют те же обозначения полей, что и файл конфигурации *wpa_supplicant*, так что, поняв принцип редактирования этого файла, вы поймете и принципы работы с графическими помощниками.

Для настройки авторизации PEAP MSCHAP V2 создайте файл **/etc/wpa_supplicant.conf** и добавьте в него запись вида

```
network={
    ssid="lochi7"
    scan_ssid=1
    key_mgmt=WPA-EAP
    eap=PEAP
    identity="kub"
    password="password"
    phase2="auth=MSCHAPV2"
}
```

Атрибут ssid содержит значение SSID беспроводной сети. Строка scan_ssid=1 необходима только в том случае, если в вашей сети запрещена широковещательная передача SSID. Значением атрибута key_mgmt должен быть список протоколов управления ключами (элементы списка разделяются пробелом). Помимо используемого нами протокола WPA-EAP, поддерживаются значения WPA-PSK, IEEE8021X и NONE (для отключения WPA). Если вы хотите явным образом указать алгоритм шифрования данных, можете указать атрибуты pairwise (для двусторонней передачи данных) и group (для широковещательной передачи). Значениями этих атрибутов являются списки имен протоколов шифрования. По умолчанию атрибуту pairwise присвоено значение **TKIP CCMP**, а атрибуту group – значение **CCMP TKIP WEP104 WEP40**. Если вы хотите, например, запретить использование любых протоколов, кроме AES, в процессе двусторонней передачи данных, добавьте в конфигурационную запись строку:

```
pairwise=CCMP
```

Атрибут eap позволяет указать список поддерживаемых методов аутентификации EAP. Мы присваиваем ему значение PEAP. В принципе, атрибут eap можно опустить, так как по умолчанию в *wpa_supplicant* включается поддержка всех методов аутентификации, поддерживаемых выбранным протоколом. Атрибут identity содержит имя пользователя для аутентификации по протоколу MS CHAP V2, а атрибут password – пароль. Наконец, атрибут phase2 указывает, что в безопасном туннеле, созданном по протоколу PEAP, должен использоваться протокол аутентификации MS CHAP V2. После внесения изменений в файл **wpa_supplicant.conf**, во избежание появления довольно странных ошибок, следует перезапустить сетевой сервис (а не только демон *wpa_supplicant*). Не забудьте (при необходимости) перенастроить точку доступа, и если вы все сделали правильно, вы увидите сообщения сервера FreeRADIUS об успешном прохождении процедуры аутентификации PEAP MS CHAP V2.

Для аутентификации по протоколу EAP-TLS на клиентской машине потребуется установить клиентский сертификат, корневой сертификат, удостоверяющий подпись клиентского и серверного сертификатов, и файл секретного ключа клиентского сертификата. Методы генерации всех этих файлов были подробно рассмотрены нами в предыдущей статье (LXF96), и сейчас нам остается выполнить лишь одно дополнительное преобразование. Напомню, что мы создавали файлы сертификатов в формате DER, тогда как демон *wpa_supplicant* и его друзья предпочитают сертификаты в формате PEM. Преобразовать клиентский сертификат из формата DER в формат PEM можно с помощью команды:

```
openssl x509 -in user1.crt -out user1.pem
```

Точно также следует преобразовать и корневой сертификат.

При аутентификации по протоколу EAP-TLS запись network {...} в файле **wpa_supplicant** должна принять вид

```
network={
        ssid="lochi7"
        key_mgmt=WPA-EAP
        eap=TLS
        identity="user1"
        ca_cert="/usr/local/etc/certs/myroot_ca.pem"
        client_cert="/usr/local/etc/certs/user1.pem"
        private_key="/usr/local/etc/certs/user1.key"
        private_key_passwd="password"
}
```

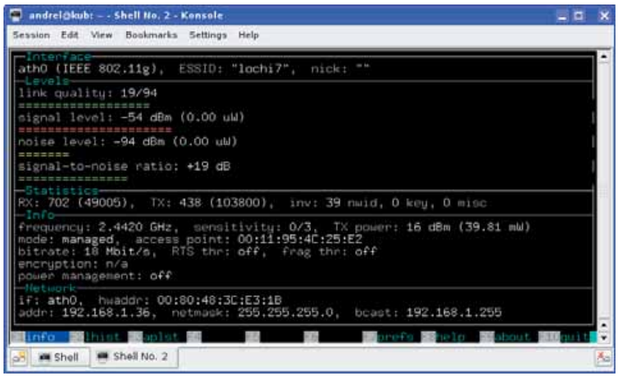
Теперь атрибут eap указывает метод авторизации TLS. Атрибут identity содержит имя пользователя, с которым связан клиентский сертификат. Как отмечалось в предыдущей статье, мы можем настроить сервер FreeRADIUS таким образом, чтобы он не проверял это значение. Значением атрибута ca_cert должен быть путь к файлу корневого сертификата, значением атрибута client_cert – путь к файлу клиентского сертификата, а значением атрибута private_key – путь к файлу секретного ключа клиента. Пароль доступа к секретному ключу хранится в атрибуте private_key_passwd. На этом настройка клиента для авторизации по протоколу EAP-TLS закончена. Как и при настройке MS CHAP V2, нам следует перезапустить сетевую систему.

Мы не коснулись еще одного важного аспекта настройки беспроводного клиента – поддержки нескольких сетей. Настроить *wpa_supplicant* для работы с несколькими сетями очень просто – достаточно добавить соответствующие записи network{...} в файл **/etc/ wpa_supplicant.conf**. В большинстве случаев подключение к доступной сети будет выполняться автоматически.

Вспомогательные утилиты

Проверить параметры подключения к беспроводной сети можно с помощью утилиты *wavemon* (**Рис. 2**). Эта консольная утилита позволяет отслеживать в реальном времени такие параметры беспроводного

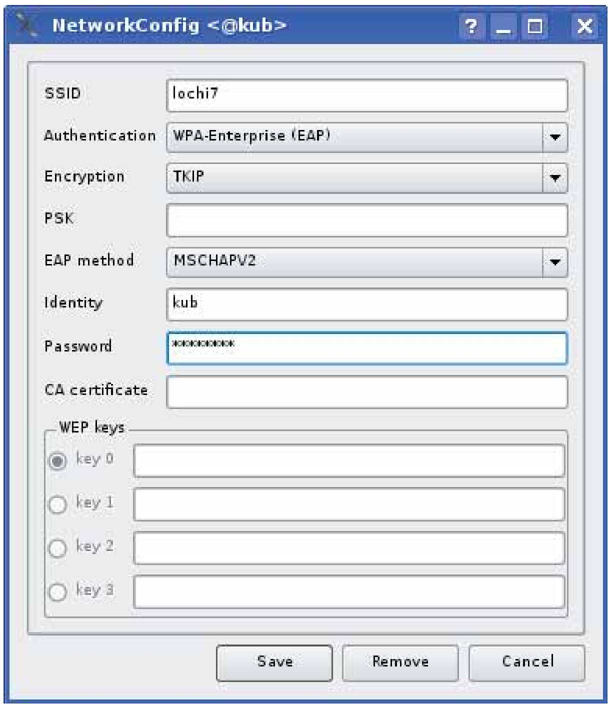
соединения, как скорость передачи, количество переданных пакетов, трафик, уровень сигнала и отношение сигнал/шум.



› Рис. 2. Проверка работы беспроводной сети с помощью *wavemon*.

Было бы странно, если бы в эпоху безалкогольного шампанского и электрических зубных щеток не существовало GUI-утилит для настройки *wpa_supplicant*. С одной из таких утилит – *nm-applet* – мы уже познакомились. Если вы почему-либо не можете использовать *NetworkManager* для настройки WPA-соединения, к вашим услугам две другие графические утилиты – *wpa_gui* и *KWlan*.

Утилита *wpa_gui* предоставляет в распоряжение пользователя графический интерфейс (на базе *Qt*) и довольно удобные возможности настройки протоколов, методов и параметров аутентификации в беспроводных сетях (**Рис. 3**), а также средство сканирования эфира в поисках точек доступа. С помощью *wpa_gui* можно настраивать параметры аутентификации в нескольких беспроводных сетях. Переключение между сетями выполняется при выборе имени новой сети в раскрывающемся списке Network.



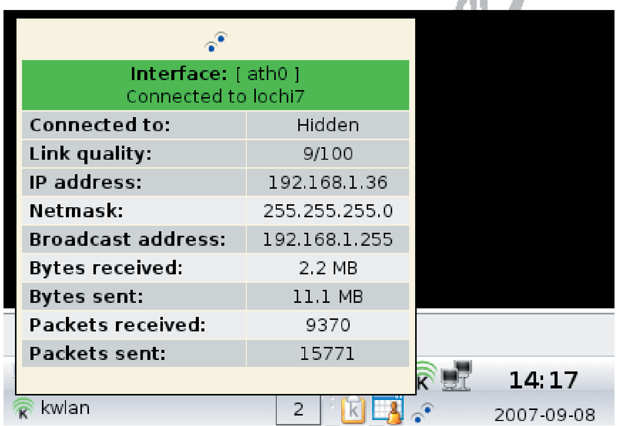
› Рис. 3. Настройка *wpa_supplicant* с помощью *wpa_gui*.

Базирующийся на *wpa_gui* сетевой менеджер *KWlan* умеет управлять не только беспроводными, но и кабельными соединениями, а также подключениями dial-up. В случае работы с беспроводными соединениями *KWlan* выступает в роли графической оболочки для демона *wpa_supplicant*. Возможности управления настройками WPA у *KWlan* те же, что и у *wpa_gui* (настройка протоколов и методов аутентификации, создание нескольких профилей для подключения к разным сетям), но интерфейс *KWlan*, на мой взгляд, удобнее (**Рис. 4**).



› Рис. 4. Настройка беспроводного соединения с помощью *KWlan*.

Помимо настроек аутентификации WPA, *KWlan* поддерживает настройки IP-адреса, адресов шлюза, DNS и других параметров соединения для каждого профиля. *KWlan* может отображать текущее состояние сетевого соединения (**Рис. 5**) и запускать/останавливать демон *wpa_supplicant*; впрочем, последняя функция пока что работает не очень надежно.



› Рис. 5. *KWlan*: ура, беспроводное подключение работает!

Подводя итоги, можно утверждать, что по простоте настройки аутентификации WPA Linux приближается к Windows (чего, увы, все еще нельзя сказать о спектре поддерживаемых беспроводных карт). В этой статье были рассмотрены не все возможности и инструменты настройки WPA Enterprise на платформе Linux, но теперь, когда общая картина вам ясна, использование любого специфического инструмента не должно быть для вас сложным делом. LXF

В помощь методисту

Сегодня **Галина Пожарина** расскажет о свободном программном обеспечении для ведения образовательного процесса.



Программное обеспечение учебном процессе может применяться как:

- базовое для формирования умений и навыков у учащихся (в соответствии с образовательным стандартом);
- обеспечивающее функционирование учебной компьютерной лаборатории (сетевые взаимодействия, антивирусная защита, политика доступа пользователей);
- обеспечивающее работу ТСО и периферийного оборудования (интерактивная доска, проектор, принтеры и пр.);
- обеспечение работы администрации учебного заведения (почта, планировщик задач, бухгалтерские программы, кадровые программы, документооборот, мониторинг учебного процесса).

Стандартные подходы к формированию сборников программного обеспечения базируются на классификации программного обеспечения по функциональным группам (офис, почтовые, антивирусы и пр.). Специфика создания подобных сборников для учреждений образования заключается в том, что образовательный процесс подразделяется с точки зрения использования информационных технологий на несколько потоков (циклов).

При подборе собственно учебного программного обеспечения, основываясь на перечисленных в образовательных стандартах перечнях формируемых умений и навыков у учащихся, видится целесообразным сопоставить их с функциональными возможностями подбираемых компьютерных программ. Перечень программного обеспечения для реализации учебной программы по предмету «информатика» определяется государственным образовательным стандартов и содержится в примерных учебных программах. Эта нормативная документация находится в открытом доступе на портале Единое окно (http://window.edu.ru/window)

Предмет «информатика» не включает:

- использование репозиториев в рамках других предметов (свободный поиск в Интернете, электронные энциклопедии, электронные учебники);
- использование преподавателями мультимедийных демонстрационных приложений (презентаций, фильмов, аудиозаписей и пр.);
- контроль качества образования вне зависимости от изучаемой дисциплины (тестирования и экзамены);
- использование компьютера в качестве составляющей профильного лабораторного оборудования (физика, электроника и пр.);
- использование компьютера в работе над проектами (внутри одной дисциплины или междисциплинарными).

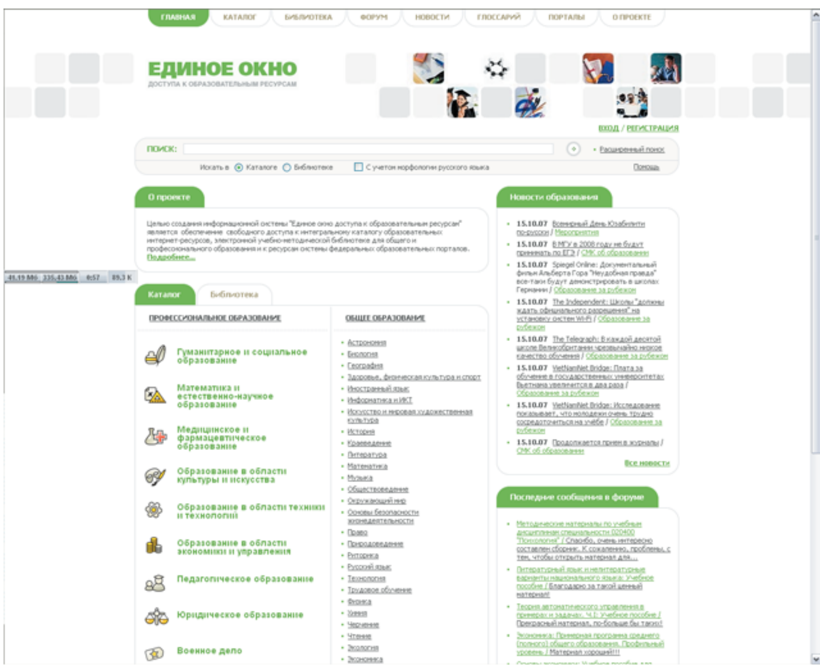
Программное обеспечение, необходимое для реализации перечисленных задач подразделяется на:

- среды для формирования умений и навыков, таких как умение печатать, рисовать, обрабатывать фотографии , видео или звук, пользоваться электронной почтой и пр.;
- репозитории информации – электронные учебники, учебные порталы и пр. как источники информации);
- контролирующие среды – контрольно-измерительные материалы для оценки качества получаемого образования: тесты, опросы);
- вспомогательные прикладные программы и модули, такие как конвертеры курса валют, программы-переводчики, словари, географические и астрономические карты, часы, таблица Менделеева и пр.;

За счет насыщения продуктами из сектора свободного программного обеспечения возможно повышение качества образования. Свободные программы зачастую пишутся студентами высших учебных заведений в качестве курсовых и дипломных работ. Их функции не только понятны, но и могут быть адекватно сравнимы с компонентами образовательного стандарта, а набор формирует «инструментарий учащегося», который можно профилировать в зависимости от предмета изучения или уровня обучения.

Репозитории информации, имеющие актуальность для образования состоят из следующих ресурсов:

- Предметные и межпредметные порталы и сетевые энциклопедии.



С позиции доступности ресурсов – наиболее доступные. Использование программного обеспечения для просмотра, копирования либо удаленной работы с подобными ресурсам, имеющими удобный web-интерфейс, сводится к использованию, в первую очередь, браузеров.

» Энциклопедии и учебники для LAN и автономной работы

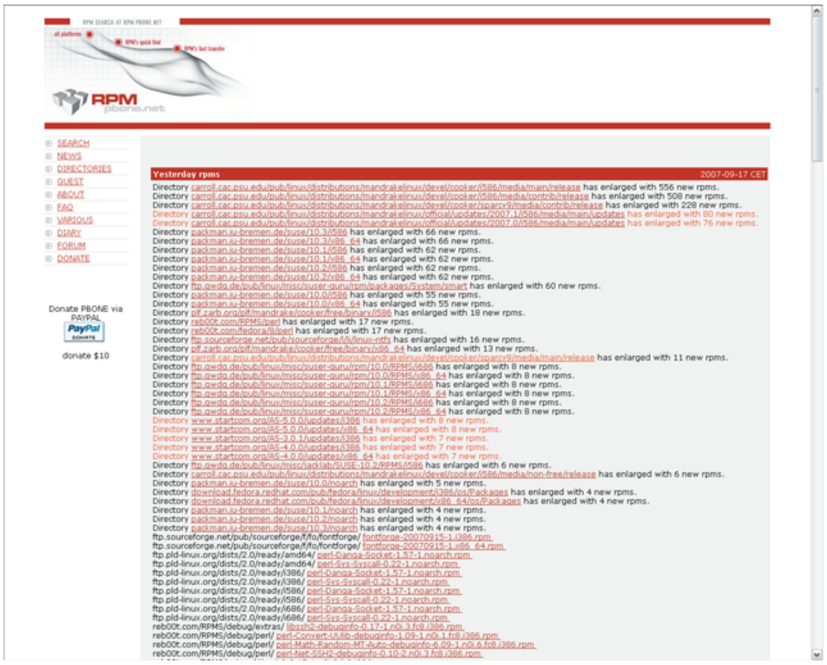
Работа с этим классом программ также в большинстве случаев связана с использованием web-интерфейса и браузеров. Технические требования, указанные в сопроводительной документации к учебнику (диску) могут служить основанием для принятия решения о возможности его установки на имеющееся в учебном заведении оборудование.

Контрольно-измерительная среда может быть внедрена в портальную систему (учебный портал), быть автономной или сетевой. Результаты опросов могут накапливаться, обрабатываться в виде сводок, экспортироваться.

» тестирование – способ «по умолчанию». Тестирующая система должна включать несколько видов тестов. Наиболее распространенные из них – выборки один из многих, несколько из многих, установление последовательности, установка соответствий, дополнение. Три первые из указанных видов тестов могут быть оценены автоматически, а следовательно лежат в основе независимой экспертной оценки. В качестве дополнительных экспертных параметров можно добавить время выполнения тестового задания, последовательность выполнения заданий теста (выполнение от первого к последнему, пропуск заданий и возможные возвраты к невыполненным заданиям с целью обдумать варианты ответа или внести исправления.

» Мониторинг «прилежности» – в сетевых портальных системах возможна оценка времени пребывания пользователя в сети, виды взаимодействий с сетевыми ресурсами. Количественными показателями могут стать также:

- публикации в сетевых ресурсах,
- активность в форумах,
- рейтинг авторского ресурса (графической работы, программы) и пр.



При подборе программного обеспечения для ТСО и периферийного оборудования , необходимо принимать во внимание, что рынок подобного оборудования достаточно обширен. Наличие репозиториев драйверов периферийных устройств дает возможность подключить имеющееся оборудование и активно его использовать. LXF

Школьный софт:

как поменять коня на переправе

«У самовара» с Александром Поносовым



Как следует из заявления Л.Реймана, до конца 2009 года следует ожидать внедрение в систему образования отечественного пакета свободных программ. Если не брать в расчет политическую составляющую этого мероприятия, то ничего сверхестественного не происходит, российскую школу реформами удивить трудно. Можно рассматривать переход на альтернативные программы как один их необходимых элементов модернизации. Другой вопрос, насколько качественно все будет организовано, удастся ли избежать сбоев в работе и прочих негативных последствий. Учитывая традиционную неспешность и обстоятельность родного ведомства, для первого разговора пригласим тех, кто уже на линейке готовности, российских разработчиков программного обеспечения.

«У самовара» в этом номере:

» **Алексей Смирнов**, генеральный директор компании ALT Linux



» **Алексей Новодворский**, заместитель генерального директора ALT Linux



» **Олег Садов**, технический директор ОАО Линукс Инк.



Итак, о грядущем «триумфальном шествии» свободного софта.

Фото: www.opensource-forum.ru; www.cnews.ru; www.cio-world.ru

Александр Поносов: **На сегодняшний момент уже имеется опыт внедрения СПО в образование, как успешный, так и неудачный. Поскольку сейчас речь идет о широкомасштабном переходе, поневоле вспоминается поставка компьютеров в Южном федеральном округе в 2003-2004 годах. Результат: пиар-акция Microsoft «Школы Волгограда отказываются от Linux в пользу платформы Windows». В чем, на Ваш взгляд, причины неудачи эксперимента? Какие уроки можно извлечь из этой истории?**

Олег Садов: Неготовность инфраструктуры поддержки и обучения. Наконец, неготовность самого решения и методической поддержки для него. И это не упрек, а скорее констатация фактов. Вся поставка происходила в крайней спешке, без какой либо предварительной подготовки. Мы тогда трезво оценили обстановку и просто отказались от участия в этой компании.

Такие масштабные внедрения надо тщательно готовить технологически, методически и организационно.

Алексей Новодворский: В ЮФО в 2003-2004 не было внедрения, была поставка компьютеров с предустановленным ALT Linux Junior. После проведения конкурса руководство программы «Дети России» сменилось, и мы оказались отрезаны от всякой информации по поставкам, а поставщик компьютеров, вопреки обязательствам, стал оказывать поддержку сам. Даже областные отделы образования не сообщали нам, в какие школы поставлены компьютеры с ALT Linux. Мы могли помочь только тем, кто сам обращался к нам.

Несмотря на это, Junior все же прижился в школах, хоть и в немногих. Об этом говорят, например, заявки школ на ПО: http://www.svobodanews.ru/Popup/2007/05/08/20070508204158180.html (таблица внизу), где он – единственный представитель свободного ПО, хотя цифры нельзя признать удовлетворительными. Нас многому научила та история. Возможно, главный вывод таков: если мы хотим успеха, то надо работать с государством, с системой образования и непосредственно с учителями. К счастью, у нас много учителей-энтузиастов СПО. Но гораздо больше тех, то хотел и мог бы использовать его, но скован по рукам и ногам нормативкой, отсутствием учебных и методических пособий. Надо ликвидировать эти препятствия, иначе самые лучшие программы не смогут быть востребованы.

АП: **По вашему мнению, как должен быть организован переход системы образования на СПО, чтобы получить гарантированный положительный результат?**

ОС: Процесс миграции должен начинаться в небольшом числе пилотных зон внедрения. Главное – наличие заинтересованных людей в этих пилотных зонах, поскольку без такой заинтересованности проект изначально обречен на провал: у учителей и так масса дел, выделенных кадров на поддержку ИТ-инфраструктуры школы в штатном расписании не предусмотрено, да пользоваться проще всего тем, к чему давно привык.

Алексей Смирнов: Немедленный тотальный и принудительный перевод всех школ на Linux был бы серьезной ошибкой, дискредитирующей само начинание. Проводя предварительные переговоры с потенциальными поставщиками, Мининформсвязи разделило стратегическую задачу перехода на свободное ПО и тактическую задачу срочной легализации уже установленных на школьных компьютерах программ. При этом была анонсирована готовность приступить к решению стратегической задачи немедленно, запустив подготовку адаптированного для нужд школ дистрибутива Linux и организовав внедрение в пилотном регионе. Возникает 2-3 летний переходный период, в течение которого нужно без потрясений перейти к широкому использованию свободных программ, решив такие проблемы, как подготовка учителей, разворачивание системы техподдержки, переход на многоплатформенные прикладные программы. Если этого не сделать, школы столкнутся с невозможностью получить квалифицированную поддержку по новой для преподавателей системе и с невозможностью запускать привычные обучающие программы, которые разработаны так, что работают только под Windows.

продолжение на стр. 106

Linux Свято-Алексеевский

Перевести школу на свободное ПО всегда непросто. А если школа эта – частная, с особым подходом и требованиями к качеству обучения? Непосредственный участник одного из таких переходов **Михаил Журухин** готов поделиться с вами своим опытом.

«Лучше день потерять, зато потом за пять минут долететь»
Крылья, Ноги и Хвосты

Велико значение тех знаний и умений, которые закладываются с детства, со школы. Это особенно касается навыков, которые позднее практически не углубляются, но активно используются в профессиональной деятельности. Речь идет об изучении информационных технологий в школе и основах компьютерной грамотности. Не является секретом то, что большая часть практических навыков работы с компьютером остается еще со школьной скамьи. Поэтому особенно важно правильно заложить основы навыков работы с компьютерами на уроках информатики в школе.

НОУ Свято-Алексеевская школа (г. Екатеринбург) – одно из таких учебных заведений, где преподаватели серьезно подходят к формированию профессиональных навыков учащихся, в том числе и в области информатики. Это небольшая частная православная школа с классической программой обучения. Сравнительно маленькие классы (от 3–5 до 15 человек), по одному в каждой параллели, позволяют реализовать индивидуальный подход к учащимся.

Обучение информационно-компьютерным технологиям (ИКТ) ведет опытный преподаватель, инженер (выпускник радиотехничес-





› Отец Роман (Важенин)

кого факультета УГТУ-УПИ) и священник – о. Роман (Важенин). Он же является лаборантом, т.е. системным администратором школы. В основе программы лежит учебник Н. Д. Угриновича «Информатика. Базовый курс» и его же методическое пособие «Преподавание курса "Информатика и ИКТ" в основной и старшей школе». Предмет преподается углубленно, с седьмого класса.

Излишне говорить, что изучение ИКТ немыслимо без соответствующего ПО. И здесь возникают сразу две проблемы: во-первых, небольшая частная школа не может позволить себе приобретение дорогостоящих программ и пакетов (а после «дела Поносова» покупка лицензионного ПО стала необходимостью), а во-вторых, с помощью коммерческого ПО для Windows трудно привить учащимся понимание процессов, протекающих в компьютере, поскольку их подробности обычно тщательно замаскированы и скрыты под глянцевым интерфейсом пользователя.

Выход из этой ситуации очевиден – использование в обучении ИКТ ОС Linux и свободного программного обеспечения (СПО). Мало того, оказалось, что Linux и СПО прекрасно заменяют коммерческие программы и дают гораздо большие возможности в преподавании и обучении, чем можно было предположить. Разнообразие СПО дает возможность абстрагироваться от конкретных продуктов и вести обучение информационным технологиям, а не пользованию определенной программой определенной версии.

Мысль использовать ОС Linux и СПО в учебном процессе появилась с того момента, когда стало ясно, что только с помощью этой системы можно наглядно объяснить учащимся все тонкости информационных процессов. Кроме того, выяснилось, что для Linux существует много хорошего образовательного ПО, которое ко всему прочему является абсолютно свободным и бесплатным. Желание расширить кругозор в информатике, привить учащимся свободный стиль мышления, рассказать о проблемах, которые существуют в мире ИТ (патенты, лицензии), показать действительно реальную альтернативу коммерческим решениям тоже стало своего рода толчком к подготовке перехода на Linux. Наконец, в программе обучения и учебниках Н. Угриновича обучение построено на трех ОС: Windows, Linux и Mac OS X, причем к методическому пособию прилагается два диска: ПО для обучения под Windows и ALT Linux с обучающим ПО (правда, это сравнительно устаревший и урезанный вариант дистрибутива). В конце методички приведено краткое руководство по установке и использованию Linux.

С самого начала было решено перевести все обучение на Linux, без использования коммерческих (платных) программ, и это условие было выполнено. Следует, однако, иметь в виду, что в данной школе пойти на этот шаг было намного проще, чем в других, так как здесь есть технические специалисты и отсутствует зависимость от вышестоящих организаций. В переводе школы на Linux и СПО активно участвовала команда проекта Open-World, сайт которой находится по адресу: http://www.open-world.org.ru/.

Процесс перехода

Парк компьютеров в кабинете информатики насчитывает 10 рабочих машин (от AMD Duron 750 МГц и до Celeron D 2.4 ГГц, объем оперативной памяти – в пределах от 128 до 256 МБ) и один сервер. Все они работали под управлением Microsoft Windows 2000.

На первом этапе было решено установить свободные приложения для Windows: в первую очередь, офисный пакет *OpenOffice.org* и браузер *Mozilla Firefox*. Сервер был переведен на CentOS (свободный клон Red Hat Enterprise Linux) с поддержкой следующих служб: *Apache* (web-сервер), *MySQL* (СУБД), PHP (язык программирования сценариев), CMS-система *Joomla!*, *Squid* (прокси-сервер), *vsftpd* (FTP-сервер) и *GuardDog* (брандмауэр). Преподаватель информатики настроил их все самостоятельно, используя в качестве универсального учебника и справочника книгу Билла Болла, Хойта Даффа и др. «RedHat Linux 8/9. Настольная книга пользователя» («ДиаСофт», 2004, 928 стр.).

Подборка открытого ПО от Свято-Алексеевской школы

- Клавиатурный тренажер: ***TuxType2***
- Обучающие игры: ***GCompris***

- Файловый менеджер: ***Konqueror, Thunar (Xfce)***
- Форматирование дискет: ***KFloppy***
- Архиватор: ***File-Roller***
- Антивирус: ***ClamAV*** **с графическим интерфейсом** ***KlamAV***

- Текстовый процессор: ***OpenOffice.org Writer***
- Электронный калькулятор: ***KCalc***
- Базы данных: ***OpenOffice.org Base***
- Звуковой редактор: ***Audacity***

- Электронные таблицы: ***OpenOffice.org Calc***
- Растровый графический редактор: ***TuxPaint***, ***GIMP***, ***Google Picasa***
- Векторный графический редактор: ***OpenOffice.org Draw***
- Создание презентаций: ***OpenOffice.org Impress***

- ПО для знакомства с кодировками: ***KCharSelect***
- Редактор формул: ***OpenOffice.org Math***
- Англо-русский словарь: ***KSocrat***
- Словарь синонимов: ***KThesaurus***

- Система объектно-ориентированного программирования: ***Gambas***

- Web-браузер: **Mozilla Firefox + расширения**
- Почтовый клиент: ***Evolution***
- Обмен мгновенными сообщениями: ***Gaim***
- Web-разработка: ***BlueFish***, ***NVU***
- Межсетевой экран: ***GuardDog***

- Просмотрщик математических 3D-функций (стереометрия): ***Zhu3D***
- Планетарий (астрономия): ***Kstars***, ***Celestia***
- Также активно используются программы из пакета KDEdu (география, химия, геометрия).

- Анимация: ***Gimp*** **(gif-ролики), Macromedia Flash в Wine**
- Запись компакт-дисков: ***K3b***

Информацию об этих и многих других образовательных программах можно найти по адресу: http://rus-linux.net/po.php?name=po/po-10-1.koi.

Затем пришел черед и рабочих компьютеров пользователей. Сложность перехода заключалась в том, что в распоряжении не было специального дистрибутива для школ, учитывающего отечественные реалии и образовательные программы. Разбираться с такими специфичными вещами, как Edubuntu, желания не было, так как работать нужно было сразу. Поэтому пришлось идти по пути наименьшего сопротивления и выбирать тот дистрибутив, который был известен лучше других, а также прост и удобен в использовании. Вариантов было немного: Novell SLED 10, CentOS и Mandriva Linux. Первый из них отпал как неприемлемый из-за скорости работы (выяснилось, что это достаточно тяжелый дистрибутив и к тому же не совсем свободный). Серверный CentOS по своей природе не слишком приспособлен для использования на рабочем столе; кроме того, для него сложно найти образовательные приложения. Так в качестве ОС была выбрана Mandriva Linux 2007: хорошо знакомый дистрибутив, удобный в настройке и использовании и имеющий большой репозиторий с образовательным ПО.

На следующем этапе была выбрана тестовая машина, на которую (после резервного копирования необходимых данных) и установили Mandriva Linux 2007 Free. Мы настроили систему, установили нужные приложения и убедились, что все работает как надо. Через некоторое время был составлен примерный план перехода, зафиксированы его основные стадии. Чтобы ускорить развертывание, установка дистрибутива на рабочие машины производилась по сети с сервера (через FTP).

В настройке свежеустановленной Mandriva очень помогла гордость дистрибутива – комплект графических утилит-мастеров под общим названием Mandriva Control Center. С его помощью была сконфигурирована сеть, подключены локальные репозитории (опять же доступные по FTP) и доустановлено необходимое ПО. Пришлось немного повозиться с настройкой оборудования: принтером и МФУ от Hewlett-Packard, видеокартами NVIDIA и др.

Вторая серьезная проблема – производительность при 128 МБ ОЗУ. Особенно остро она стоит для таких тяжеловесов, как *OpenOffice.org*. Проблема частично решилась с переходом на новую версию *OOo* – 2.2, частично – использованием облегченной версии Mandriva Linux – SAM Linux. Этот дистрибутив, предлагающий легкий рабочий стол Xfce4, показал высокую скорость работы, особенно с *OpenOffice.org*. Но в нем есть серьезные проблемы: в основном англоязычный интерфейс, неполная совместимость с Mandriva на уровне пакетов, сырой инсталлятор, отсутствие Центра управления. В результате было решено, что дистрибутив требует доработки, поэтому он был оставлен только на самой слабой машине.

Третья проблема: курс Угриновича предусматривает программирование на Visial Basic, Delphi, знакомство с «Компасом». Замена *Visial Basic* приведена на страницах учебника и методички – это *Gambas*. Однако он не полностью совместим с *Visial Basic* (например, у этой программной среды оказались некоторые свои особенности вывода графических модулей), а упражнения специально для *Gambas* разработаны не были. Проблема была решена переработкой практических задач *Visual Basic* для *Gambas*. Старую версию среды, включенную в состав дистрибутива, было решено заменить другой сборкой, доступной по адресу: **ftp://ftp.open-world.org.ru/pub/gambas**. От *Delphi* пришлось отказаться, в том числе и из-за нехватки учебного времени.

С Компасом ситуация сложнее. Вариантов три: отказаться от его использования в пользу *QCad*, не рассматривать совсем и попытаться запустить в Wine@Etersoft. Думается, что многие из этих проблем будут решены с выходом в свет Mandriva Education Linux.

Свои особенности нашлись и у «Открытого Офиса», однако, выручила полная русскоязычная документация: Чернова Д., Еременко А. и др. «Руководство пользователя OpenOffice.org 2 (+CD)», BHV, 2007 г. В ней рассмотрены все возможности пакета *OpenOffice.org* 2: работа с текстовыми документами в *OpenOffice.org Writer*, создание электронных таблиц в *OpenOffice.org Calc*, подготовка и демонстрация презентаций в *OpenOffice.org Impress*, создание математических формул в *OpenOffice.org Math*, подготовка рисунков в *OpenOffice.org Draw*, основы работы с базами данных в *OpenOffice.org Base* и так далее.



Были и некоторые проблемы технического характера: не работали *Tuxtype2* на русском (проблема была в переключателе раскладки), *Kxneur* (аналог *Punto Switcher*). Были сложности и с traceroute.

Первые впечатления

Настроить компьютеры – треть дела, нужно еще учить детей. С детьми легче:

– Вы поставили новый Windows?

– Нет, это другая ОС, называется Linux.

– А она лучше Vista?

– Лучше – понятие относительное, но новее – это точно.

– Ура! Нам новый Windows поставили! Mandriva называется!

Впрочем, поначалу среди детей было несколько противников перехода, особенно среди одиннадцатиклассников. «Сложно и немодно», – говорили они. После «копания в настройках» недовольных поубавилось – одна прозрачность панелей чего стоит. А после демонстрации Mandriva 2007 с «кубиком» на столе появилась целая куча дисков с просьбой «записать Linux».

Кроме старших классов, в Свято-Алексеевской школе знакомят с компьютерными технологиями и совсем маленьких: 1–3 классы. Для них разработана специальная программа, позволяющая в игровой форме ознакомиться с возможностями компьютера и информационными технологиями. Это *GCompris* (**http://gcompris.net/-ru-**). А еще детям очень нравится *TuxPaint* (красивый и удобный графический редактор), они рисуют в *GIMP* и играют в *SuperTux* и вообще во все игры, которые находят.

Независимость – тема, которая сквозит в каждом примере развертывания свободного ПО в школах. Свято-Алексеевская школа – частная, и по определению имеет больше свободы выбора учебных планов и ИКТ, чем государственные общеобразовательные школы РФ. В перспективе здесь планируется раздавать домашнее задание ученикам на LiveCD, а также проводить компьютерное тестирование по всем предметам (с помощью *iTest* и *KEduca*), организовать удаленное администрирование, демонстрировать основные сетевые технологии и углубить изучение HTML, внедрить в образовательный процесс *Google Earth* и автоматизировать организацию учебного процесса (*SchoolTool*). LXF

GNU/Linux заглядывает в ВУЗы

Многие говорят, что свободное ПО и высшее образование просто созданы друг для друга, но далеко не все знают, как организовать их первую встречу. **Александр Сергеевич Казанцев**, успешно продвигающий Linux в родному институте и за его пределами, любезно согласился ответить на вопросы **Евгения Балдина**.

Александр Сергеевич Казанцев



› **Рис. 1. Александр Казанцев собственной персоной.**



› **Рис. 2. Александр Казанцев и Михаил Давыдов – web-гуру института.**

Наличие тотального высшего образования способствует ускоренному развитию общества. Почти все грамотное население России знает про МГУ; есть подмножество, которое знает, что НГУ ничуть не хуже. Но даже если очень захочется, все просто физически не смогут обучаться только там, а обучать надо именно всех. Большинство образовывается в небольших ВУЗах, и в этих ВУЗах роль личности может оказаться определяющей.

Александр Сергеевич Казанцев (он же akdengi на ЛинуксФоруме и в Русском сообществе *OpenOffice.org*), молодой кандидат технических наук и доцент (пока без диплома) кафедры «Автоматизированные системы управления» Глазовского инженерно-экономического института (филиала) ГОУ ВПО Ижевский государственный технический университет, выбирает GNU/Linux.

Linux Format: **Кого готовит ваша кафедра?**

Александр С. Казанцев: Мы готовим специалистов-компьютерщиков широкого профиля. Студенты заканчивают специальность 230102, Автоматизированные системы обработки информации и управления, а вот кто из них в результате получается – это вопрос.

Часть работает в направлении САПР и АСУТП, часть в области бухгалтерских программ. Есть выпускники, работающие компьютерщиками в банках и в пенсионном фонде, то есть ВУЗ поставляет специалистов для разных областей. А раз работодатели сами идут к нам просить студентов[1], значит, выпускаем тех, кого надо, благо программа содержит всего понемногу: и программирование, и САПР, и проектирование систем, а также математические дисциплины, дающие для этого базу.

LXF: **Что используется в процессе обучения?**

АСК: Компьютеры, компьютеры и еще раз компьютеры. Ну, еще проектор для демонстраций. Плюс применяем сеть 1-Wire для подключения внешних управляемых систем. Две лаборатории – моя гордость, одна с «железом», где можно пощупать различные части ЭВМ, вторая – с пятью компьютерами (пока) и проводной, а также беспроводной сетью, опять же для лучшего изучения. Плюс общие компьютерные классы института (три класса по 15 машин на GNU/Linux).

Программное обеспечение необходимо разное, так как для каждой дисциплины нужен свой программный пакет. Одних ОС в процессе обучения я рассматриваю две-три штуки.

[1] Больше всего удивил военкомат (улыбается).

LXF: По какой причине возник вопрос перехода на GNU/Linux?

АСК: Первичной причиной была задача сделать быстро восстанавливаемый компьютерный класс для изучения операционных систем, позволяющий безболезненно программировать и менять их настройки, вплоть до погружения системы в нокаут. Тогда-то (два года назад) у меня в лаборатории впервые появился мой первый GNU/Linux и виртуальные машины. Затем известное дело Поносова заставило руководство ВУЗа поднять вопрос о лицензионной чистоте программного обеспечения. Тут и пригодились мои наработки. Результат – GNU/Linux почти во всем ВУЗе.

LXF: Был ли план миграции? Как производился переход?

АСК: План заключался в постепенном переходе по максимуму на открытое ПО, с закупкой лицензионных закрытых продуктов только при крайней необходимости. Я сделал выбор дистрибутива, совместно с коллегами (за закупки и обслуживание всей техники в институте отвечают другие люди) посмотрели список используемых в учебном процессе программ, нашли им аналоги из мира Open Source, работающие под GNU/Linux, и приказом директора была запущена машина перехода. Первые дистрибутивы мы получили в феврале, а к маю-месяцу переход был закончен. На кафедре я это сделал за одну неделю.

Переход выполняли своими силами – точнее, моими и двух лаборантов, один из которых – любитель FreeBSD (ничего, кроме нее, не признает), а второй – заядлый линуксоид. В процессе работы обучили еще несколько человек, так что теперь все лаборанты могут нормально развернуть систему.

LXF: А что еще осталось доделать и почему?

АСК: Доделывать осталось еще много (по институту в целом). Связано это со сменой операционных систем на серверах (трогать их во время учебного процесса было нельзя) и полном развертывании того программного обеспечения, которое было не нужно во втором полугодии. В мою компетенцию входит перевод типографии института на GNU/Linux и развертывание нового дистрибутива в лаборатории и классах.

LXF: Какой был выбран дистрибутив GNU/Linux?

АСК: Решили остановиться на Mandriva Linux, и я не жалею об этом. Выбор шел среди дружественных пользователю дистрибутивов, которые облегчили бы переход студентов и преподавателей с Windows.

LXF: Как отреагировали на переход пользователи?

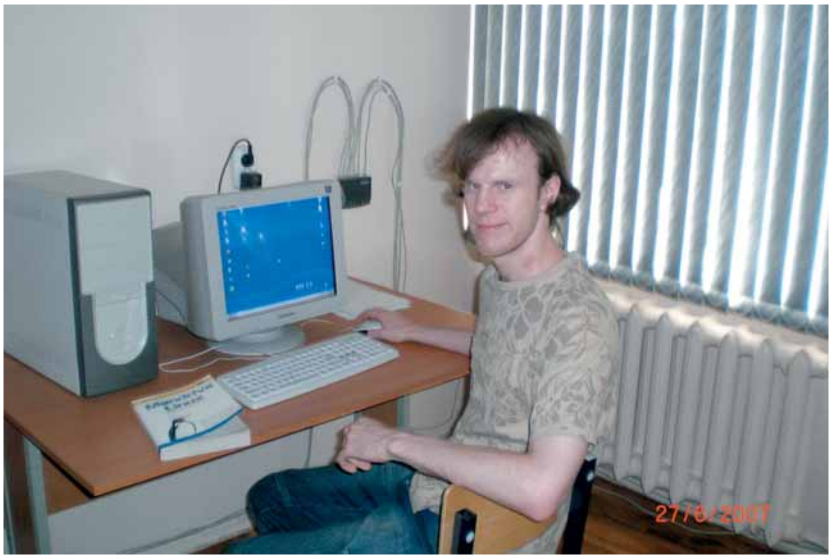
АСК: Как и следовало ожидать, пользователи отреагировали бурно отрицательно. Но внушенная им мысль о самостоятельной покупке лицензионного ПО сразу остудила обстановку. Так как большинству был необходим только «Офис», то переход прошел почти безболезненно, – кроме как в деканатах, из-за необученности государственных структур, любящих присылать различные формуляры на заполнение в виде файлов Excel с макросами и бухгалтерии.

Возникающий шум решался отсутствием Windows и поднятием вопроса о занимаемой должности. Жестко, но по-другому в сжатые сроки было не перейти. Так же я поступил у себя на кафедре, полностью удалив Windows с кафедральных машин. Благо таких эксцессов было немного, а похожесть Mandriva Linux и *OpenOffice.org* на их «несвободных коллег» сыграла свою положительную роль.

Переучивание же студентов происходило на парах по ходу занятий, а преподаватели перестраивались сами (мы, преподаватели, должны заниматься самообразованием). В результате перехода студенты разделились на два лагеря: одни меня про себя ненавидят, а другие поддерживают. Жалко, что первых пока больше. Надеюсь, что в следующем году таких уже не останется.

LXF: Насколько из-за перехода на свободное программное обеспечение изменился образовательный процесс?

АСК: Образовательный процесс ожил. По своей кафедре могу сказать, что исчезла проблема с темами курсовых и дипломных работ. Если под Windows в основном все уже написано, а некоторые темы (например, бухгалтерский учет) вообще считаются «оффтопом», то в мире свободного ПО работы хоть отбавляй. Не стало проблем с выбором языков программирования (в настоящее время используем *Free Pascal* и *C# Mono*). Пишут также на Java, C++, Ruby, Python. Появилась возможность рассматривать CRM/ERP системы и организовывать дистанционное обучение. До перехода я (да и остальные) просто не задумывались об Open Source. Обеспечена лицензионная чистота по многим дисциплинам – например, по электротехнике и микропроцессорной технике используем *KTechLab*; по компьютерной графике – *Blender* и *Inkscape*. Проще стало изучать Операционные системы, Сети ЭВМ, Моделирование систем и тому подобное.



› Рис. 3. Марат Салимзянов — линуксоид по убеждению.

LXF: Какие методические материалы сейчас используются при обучении?

АСК: Пока идет наработка материала. Используем «нарытое» в сети и из журнала Linux Format. Перерабатываем старые методические указания (с *Microsoft Office* на *OpenOffice.org*, с *MathCad* на *Maxima* и т.п.) и пишем новые. Я уже в этом году давал программирование сетевых приложений на C# в Mono. На новый учебный год (2007–2008) готовлю новые методические материалы по Операционным системам и Проектированию АСОИУ с учетом Open Source. Также буду вести новый предмет – кросс-платформенное программирование, поэтому готовлю полный методический комплект для этого предмета: лекции, практические занятия и лабораторные.

Параллельно готовлю цикл статей по использованию Mandriva Linux в образовании. Все, что напишу я сам и мои коллеги, обязательно сделаем общим достоянием на одном из ресурсов, посвященных GNU/Linux.

LXF: Раз уж упомянули Linux Format, то не могли бы вы перечислить наиболее полезные/интересные с вашей точки зрения статьи?

АСК: Особенно понравились циклы по программированию на разных языках – много полезного почерпнул (и продолжаю) из статей о Mono, Ruby и Java. Другие преподаватели (я им даю почитать свою копию журнала, который пока выписываю только я) тоже нашли массу полезного в статьях об *Inkscape*, *Blender* и *OpenOffice.org*. Для новичков оказались полезными статьи, показывающие работу с пакетами на разных дистрибутивах.

LXF: Какие новые материалы в Linux Format вам хотелось бы увидеть?

АСК: Интересно бы было прочитать о связи приложений в Linux с *OpenOffice.org* на разных языках – я считаю тему актуальной, особенно в свете отказа от несвободных офисных пакетов. Также с удовольствием бы ознакомился с программированием при помощи библиотек SDL

› **Рис. 4, 5. Институтские лаборатории.**





и OpenAL как замены DirectX в проектах. Не помешали бы обзорные статьи о математических программах (был только цикл по Maxima), программах для бухгалтерского учета, существующих CAD/CAM системах. Люди просто не знают, какие программные продукты существуют, а использование выдержек из данных материалов в дипломных и курсовых проектах позволило бы донести идеи Open Source до более широких масс.

LXF: **Чего вам сейчас не хватает для полного счастья? В смысле использования GNU/Linux в образовательных целях.**

АСК: По порядку:

» Понимания. Как со стороны студентов и сотрудников, так и со стороны руководства (очень сильное лоббирование «виндузятников», как внутривузовских, так и из высших эшелонов власти, которое может свести на «нет» все начинания).

» Образовательного дистрибутива Mandriva. Кучу вещей (например, нормально скомпилированного под Mandriva с поддержкой GTK2 и Qt Lazarus) доделываю в свободное время дома и внедряю в институтский репозиторий (надеюсь, мое начинание найдет поддержку у Mandriva.ru).

» Времени. Работы и идей много, но чтоб все успеть… Последние полгода сплю по 5-6 часов. Нужна команда единомышленников, но пока [грустно] я не вижу большого рвения со стороны коллег мне помочь.

» Хорошего САПР. Из-за него (а именно, КОМПАС-3Д) мы вынуждены были закупать «не Линукс».

LXF: **Что бы вы пожелали сообществу для того чтобы процесс внедрения GNU/Linux в образование пошел активнее?**

АСК: Нужно выходить на государственный уровень УМО, федерального агентства по образованию. Лицензировать специальности по рабочим программам с использованием GNU/Linux. Создавать образовательные ресурсы, для поддержки начинания, издавать книги и делиться опытом. А преподавателям следует заниматься положенной нам, преподавателям, работой, а не сидеть сложа руки. LXF

Школьный софт: как поменять коня на переправе (окончание, начало на стр. 100)

АП: **Думается, полного перехода на альтернативную платформу все таки не будет. Тем более, часть школ уже закупила лицензионные продукты Microsoft. Что вы думаете о моно- и гетерогенных системах в школе?**

АН: Мне представляется, что решение государства о закупке лицензий на проприетарное ПО для снятия проблемы легальности вкупе с внедрением в школы СПО весьма удачно. Какое-то время мы будем жить с двумя системами. Надеюсь, что дальше все будут определять не деньги, а стремление познакомить детей с действительно современными технологиями, привить им вкус исследованиям, к совместной работе, дать образованию инженерный импульс. Если это получится, то выбор СПО неизбежен.

ОС: Выбор – это великолепно! Самое главное – это что бы та самая «вторая» система когда-нибудь стала первой, а не лежала мертвым грузом на диске.

АС: Широкое использование многоплатформенных прикладных программ ликвидирует зависимость от поставщика какой-либо одной ОС. Мы уверены, что использование в школах двух ОС – Linux и Windows принесет несомненную пользу: учитель получает свободу выбора используемой на уроке ОС, а школьники расширят кругозор.

АП: **Что можете порекомендовать коллективам образовательных учреждений, планирующим переход на СПО, для того, чтобы он прошел безболезненно?**

ОС: Тщательно планировать процесс миграции, по возможности привлекать к этому местных специалистов.

Разумной может оказаться поэтапная модель внедрения:

» на первом этапе можно организовать использование свободного ПО для Windows (Firefox, *OpenOffice.org*, GIMP и др.)

» само знакомство с Linux можно начинать с LiveCD, который дает возможность опробовать новую систему без риска сломать то, что уже установлено на компьютере

» финальный этап – установка системы в режиме двойной загрузки, позволяющая при необходимости воспользоваться привычным программным окружением.

АС: Учитывая всю сложность ситуации, мы предлагаем трехлетний план перехода школ на свободное программное обеспечение.

» **1 этап**. Обеспечение базовой функциональности рабочего места. Прямо сейчас мы готовы выпустить специализированный «школьный» дистрибутив, который будет рассчитан на рабочее место ученика и содержать все необходимые для школьного курса программы. Дистрибутив включает многопользовательскую, многозадачную операционную систему с оконным интерфейсом пользователя, близким по «устройству» с интерфейсом Microsoft Windows, офисные приложения: текстовый процессор, электронные таблицы, систему презентаций, совместимые с форматом ODF и имеющие возможность импорта/экспорта документов в распространенные неспецифицированные форматы doc, xls, ppt, с поддержкой русского правописания и переносов.

» **2 этап**. Здесь следует сосредоточиться на решении задачи запуска в Linux имеющихся в школах обучающих и медиа-материалов, рассчитанных на ОС Windows, разработке средств простой подготовки образов загрузочных CD (LiveCD) с заданным набором прикладных программ, работающих без установки ОС. Кроме того необходимо включить программы, требуемые для организации учебного процесса: средства для обеспечения административной деятельности школы, средства для организации дистанционного обучения, средства для проведения тестирования, обучающие программы по предметам школьного курса.

» **3 этап**. Создание инфраструктуры школы и класса, включающей легко настраиваемый сервер с функциями web-сервера, сервера баз данных, сервера домена, почтового сервера, сервера мгновенных сообщений, а также шлюза в Интернет с антиспамом и антивирусом. Организация школьного класса должна обеспечивать работу ученика на любом компьютере класса с сохранением индивидуальных данных и настроек, при этом устаревшие компьютеры можно использовать в качестве терминалов.

АП: **Ваши компании будут участвовать в конкурсе на разработку пакета ПО для школ. Сроки крайне сжаты. Наверное, в самое ближайшее время можно будет познакомиться с «опытными образцами»?**

АН: Не позже следующей недели (24.09-30.09) начнем публичное тестирование.

ОС: Наш комплект поступит в школы в начале октября. Подарок ко Дню Учителя. LXF



Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

1 DVB на DVD

В Я сумел настроить тюнер DVB-T, но моя жена не хочет смотреть трансляцию на экране компьютера; нет ли способа конвертировать файлы, сохраненные из потока, чтобы воспроизвести их на нашем телевизоре через DVD-плейер?

towy71, с форумов LXF

О DVB и DVD используют два варианта видеокодеков MPEG2. DVB использует MPEG2-TS, а DVD – MPEG2-PS; соответственно, Transport Stream и Program Stream. Различаются они в основном тем, что Transport Stream предназначен для работы при ненадежной связи, типа радиоканала, поэтому имеет большую избыточность кода и коррекцию ошибок и создает файлы примерно на 30% больше. Перекодирование из MPEG2-TS в MPEG2-PS является простым и быстрым, поскольку оно затрагивает лишь эту восстановительную информацию, а не формат видео.

Существует ряд программ, пригодных для превращения DVB MPEG в DVD. Одна из самых простых, хотя и довольно медленная – *tovid* (http://tovid.wikia.com); команда *todisk* из пакета берет список видеофайлов почти любого формата и преобразует их в ISO-образ DVD. Если вы предпочитаете графический интерфейс, полезны будут *dvdstyler* (www.dvdstyler.de) и *qdvdauthor* (http://qdvdauthor.sourceforge.net). Однако для цели создания DVD из единственного MPEG2-файла они будут перебором – скрипт оболочки сделает эту работу быстрее:

```
#!/bin/sh
mplayer -dumpfile title.audio -dumpaudio $1
mplayer -dumpfile title.video -dumpvideo $1
mplex -f 8 -o title.mpg title.{audio,video}
dvdauthor -x title.xml
mkisofs -dvd-video -o title.iso dvd
```

Здесь **title.xml** содержит:

```
<dvdauthor dest="dvd">
<vmgm /><titleset><titles>
<pgc><vob file="title.mpg" /></pgc>
</titles></titleset>
</dvdauthor>
```



› Конвертация видеофайлов DVB (Freeview) в DVD проста: по сути они одинаковы.

Этот скрипт разделяет аудио- и видеопотоки, затем перекомбинирует их с данными способом, принятым для DVD – без излишеств DVB, создаст файловую структуру DVD и запишет ISO-образ. Перед прожигом ISO-образа на DVD можете протестировать его:

```
mplayer -dvd-device title.iso dvd://1
```

Для этого понадобятся *mplayer*, *mjpegtools* и *dvdauthor*; все они должны быть в репозитории вашего дистрибутива, а большинство, вероятно, уже установлено. Кроме того, если вы используете *MythTV* для записи и просмотра программ, установите модуль расширения *mytharchive* для экспорта в DVD. Это приложение может поместить несколько программ на один диск – при необходимости перекодируя их для размещения большего числа на одном носителе (получается дольше, но если вы собираетесь делать это регулярно и не хотите быть погребены под кучей дисков, то стоит им воспользоваться). Оно также предлагает на выбор стили и раскладки меню. Я сам этим постоянно пользуюсь. **НБ**

Наши эксперты

» Мы найдем эксперта по любому вопросу. Вы получите ответ на все: от проблем с установкой или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный супер-программист. Он управится со всеми вашими проблемами по части web и баз данных.



Ник Вейч
Во время, свободное от исчеркивания текстов красными чернилами, Ник возится с Linux- графикой и 3D-приложениями; он у нас отвечает за простые вопросы!



Владимир Иванов
Владимир работает с Linux уже более 10 лет, днем – создавая комплексные решения для крупных компаний, а вечером – модерируя сайт linux.org.ru и отвечая на вопросы о системном администрировании и информационной безопасности.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите нам по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

2 Кнопки-уродцы

В Я новичок в Linux: избавился от Windows XP и теперь сижу в PCLinuxOS 2007 на моем Fujitsu Siemens Amilo xi 1546. При использовании *Firefox* радиокнопки на web-страницах выглядят уродливо, а не так гладко и округло, как в *Internet Explorer*. Есть ли решение, способное сделать их приятнее на вид?

Порывшись в Сети, я нашел нечто о помещении изображений пары радиокнопок в папку **/Firefox/res** и правке кода в **/Firefox/res/forms.css**, но ссылки на код и изображения устарели и исчезли. Объясните новичку в Linux, как это сделать попроще?

Ник [Nick]

О По умолчанию виджеты *Firefox* и вправду выглядят угловато. Вы, вероятно, наткнулись на модуль Осмо Саломаа [Osmo Salomaa], его можно загрузить с http://users.tkk.fi/~otsaloma/art/firefox-formwidgets.tar.gz. Но чтобы избавить вас от проблем на случай, если ко времени выхода журнала »

» архив снова «сбежит в самоволку», мы включили этот файл в директорию **Magazine/Answers** на *DVD Linux Format* этого месяца. Для его установки выйдите из *Firefox*, скопируйте **firefox-form-widgets.tar.gz** с DVD в вашу домашнюю директорию, затем откройте терминал и введите

```
tar xf firefox-form-widgets.tar.gz
cd firefox-form-widgets.tar.gz
su
cat res/forms-extra.css >>/usr/lib/firefox-2.0.0.3/
res/forms.css
cp -a res/form-widgets /usr/lib/firefox-2.0.0.3/res/
exit
```

Для изменения системных файлов Вам потребуются права суперпользователя root, этот вопрос решается командой **su**. Команда **exit** снова переключит Вас на обычного пользователя, так как неразумно оставаться под правами **root** без абсолютной необходимости.

Кстати, пользователи Ubuntu имеют графический инсталлятор для этих виджетов, благодаря любезности одного из своих форумчан. Вы найдете его по ссылке **http://ubuntuforums.org/showthread.php?t=369596** **ПХ**

3 Критика обновлений

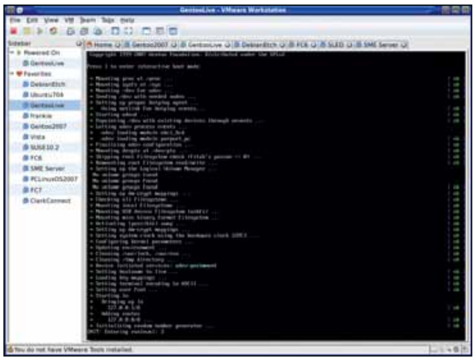
Я администратор в небольшой компании, предоставляющей Web-сервисы зарубежным филиалам, VPN-решения и другие услуги, все на базе CentOS, а также внутренние сервисы, такие как *Samba* и CUPS. Я не очень разбираюсь в обновлении Linux-серверов, но должен сделать это прямо сейчас. Паралич из-за страха все сломать не может тянуться долго – это приведет к ненадежности системы. Я читал книгу за книгой, статью за статьей. Все они упорно твердят: «Сделайте резервную копию ваших данных». Каких данных? Это же не просто архивирование домашней директории, ведь сервисы уровня предприятия охватывают всю ОС, и одни библиотеки зависят от других. Вдруг из-за обновления что-нибудь рухнет? Можно ли сделать «откат»?

Я понимаю, что основные серверные дистрибутивы тратят много времени, чтобы пакеты в их репозиториях были самодостаточными, однако есть вещи, которые никогда не сделать силами дистрибутива – например, некоторые CRM, почтовые решения третьих фирм и т.д. Наличие более одного пакета с похожей функциональностью может реально означать, что я в конечном итоге буду распутывать зависимости вручную, если что-то пойдет не так. Идеальным решением, конечно, было бы сперва применить заплатку на какой-нибудь тестовой среде. По-честному, сколько людей делают зеркала всех бесперебойно работающих служб? У меня есть резервный компьютер, но я предпочитаю его не трогать, а держать на случай, если сломается что-то другое.

Виртуализация кажется выходом из этой ситуации. Виртуализуем свое окружение, делаем снимок и применяем заплатку, а если что-то пойдет не так – откатим всю операционную систему. Но это выглядит слегка неэлегантно – все равно что менять автомобиль, когда кончился бензин.

sgb, с форумов LXF

Пример со сменой автомобиля тут не совсем уместен: откат к последнему снимку просто уберет изменения, внесенные после его снятия, аналогично функции undo – только не на одно действие, а на фиксированное время.



› Сохранение копий ваших серверов как виртуальных машин позволяет легко тестировать новое программное обеспечение без привлечения запасных компьютеров.

В серверах, критичных для бизнеса, все изменения нужно обязательно тестировать на отдельной системе перед их применением. Вы правильно мыслите насчет виртуализации, но ее надо применить в тестовой системе: так можно эффективно испытывать версии всех ваших систем на одной или двух машинах. Это дает ряд явных преимуществ. Во-первых, можно использовать один компьютер с рядом виртуальных машин на нем, который потребует не больше ресурсов, чем отдельные системные блоки для каждого из ваших серверов, за очевидным исключением дискового пространства. При необходимости обновить какую-то систему, просто загрузите виртуальную машину, примените и испытайте обновления и реплицируйте их на требуемый сервер, когда будете полностью удовлетворены надежностью работы. Если появятся проблемы, вернитесь на предыдущий снимок и попробуйте еще раз; а ваш производственный сервер будет спокойно заниматься своим делом.

Еще одно преимущество тестирования на отдельной системе проявится, когда вы будете устанавливать программы из исходных текстов. На реальной системе их собирать не нужно, поэтому на ней не нужно и держать полный комплект средств разработки. Это сокращает число пакетов, установленных на удаленном сервере, а значит, повышает его безопасность. Чтобы собрать RPM-пакеты программ для установки на производственных системах, воспользуйтесь *checkinstall* (**http://checkinstall.izto.org**). **НВ**

4 Эй, Raid!

Мы установили сервер *Apache Tomcat* с двумя дисками на 500 ГБ с программным RAID 1. Я кое-что поменял в некоторых файлах, перезагрузил сервер, чтоб их протестировать, и обнаружил, что внесенные изменения исчезли. Файлы, которые я удалял, появились снова. Я проверил свою почту и получил письма с ошибками от **mdadm**.

```
A DegradedArray event had been detected on md device
/dev/md0.
The /proc/mdstat file currently contains the following:
Personalities : [raid1]
md1 : active raid1 sda2[0] sdb2[1]
1959808 blocks [2/2] [UU]
md0 : active raid1 sda1[0]
486424000 blocks [2/1] [U_]
unused devices: <none>
```

Я делаю резервное копирование всей важной информации, но хочу по возможности спасти сервер, »

Коротко про...

Файловые системы

Различные способы хранения и организации всех ваших данных.

Файловая система – это метод хранения и организации данных на ваших запоминающих устройствах, таких как жесткие диски или DVD. Все файловые системы решают одни и те же задачи, но разными способами. Именно поэтому каждая файловая система имеет свои сильные и слабые стороны.

Много лет единственной файловой системой, используемой в Linux, была ext2 – весьма надежная, благодаря столь долгому тестированию, а также быстрая. Одна из причин ее быстроты заключается в отсутствии ведения журнала. Попросту говоря, журналирование – это регистрация изменений в файловой системе, облегчающая восстановление ее целостности после незапланированного отключения компьютера. В файловой системе ext2 выполнение **fsck** после сбоя или отключения электроэнергии может отнять много времени.

При добавлении в ext2 журналирования получилась ext3, по-прежнему предпочитаемая многими в силу ее долгой истории и множества инструментов восстановления. Две другие популярные файловые системы, предлагаемые на выбор при инсталляции во многих дистрибутивах – ReiserFS и XFS. Первая использует особый метод хранения файлов, повышающий эффективность обработки большого числа маленьких файлов – например, вашей домашней директории со всеми почтовыми файлами и кэшем браузера.

XFS использует кэширование памяти в большей степени, чем другие файловые системы. Это делает ее очень быстрой, но гораздо более уязвимой в случае сбоев в сети электропитания. Для систем, защищенных ИБП, или для ноутбуков, где батарея позволяет не бояться внезапного отключения, это хороший выбор; но будьте осторожны с ее использованием в незащищенном компьютере. Есть и другие файловые системы Linux, например, JFS от IBM и новые файловые системы – Reiser4 и ext4, но они (пока) не нашли широкого применения. [развитие файловой системы Reiser4 приостановилось из-за заключения под стражу ее разработчика, а ext4 пока находится в стадии тестирования, – *прим. пер.*]

Файловые системы Windows заслуживают упоминания благодаря их массовой распространенности. Даже если вы не используете Windows, вы можете использовать USB-брелки, цифровые камеры и MP3-плейеры. Все они используют файловую систему Windows FAT, которую Linux может без проблем читать и писать. Более поздняя файловая система NTFS, используемая в WindowsXP и Vista, в настоящее время допускает надежную запись, благодаря модулю *FUSE ntfs-3g*.

поскольку его установка была весьма специфическим и долгим процессом. Я новичок в мире администрирования Linux и не знаю, с чего начать.

Генри Анджелес [Henry Angeles]

О Содержимое **/proc/mdstat** показывает, что произошел сбой диска в массиве **md0** (**/dev/sdb1?**). Ваша машина будет продолжать работать с деградированным массивом, но с несколько сниженным быстродействием и без резерва на случай выхода из строя другого диска. Инструментов для тестирования дисков довольно много, но самый надежный вариант – заменить диск и пересобрать дисковые массивы. Конечно, автоматически будет заменен и **/dev/sdb2**, так что другой массив также потребует пересборки. К счастью, это задача простая и в значительной степени автоматизированная, но потребует времени. Правда, после замены неисправного диска вы сможете продолжить работу на компьютере, параллельно с восстановлением массивов, но при заметном замедлении дисковой подсистемы.

Будет легче, если вы сможете добавить новый диск до удаления старого, тогда можно будет сперва восстановить **md0**, а затем переключить **md1** на новый диск в любой удобный момент. Если новые диски добавляются в вашу систему как **/dev/sdc**, подключите его и перезагрузитесь. Затем создайте на нем разделы так же, как вы это делали на **sda** и **sdb**, установив тип разделов Linux Raid Autodetect. Теперь запустите эти команды с правами **root**, для удаления неисправного диска из массива и добавления нового:

```
mdadm /dev/md0 --fail /dev/sdb1 --remove /dev/sdb1
mdadm /dev/md0 --add /dev/sdc1
```

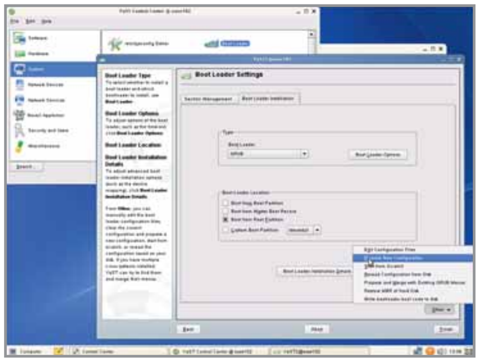
Когда новый диск добавится в массив, RAID-драйвер синхронизирует его с существующим диском. Это потребует времени; контролируйте прогресс работы по содержимому **/proc/mdstat**. По завершении процесса вы будете иметь оба массива работающими правильно, но использующими три диска, один из которых ненадежен, поэтому повторите эти команды для **md1**, **sdb2** и **sdc2**, чтобы перенести второй массив на новый диск. Теперь можно выключить компьютер и убрать неисправный диск, когда захотите, так как он уже не используется.

Разумеется, как и при любой критической операции над диском, прежде чем приступать к данной процедуре, позаботьтесь о резервной копии Ваших данных. Старый жесткий диск можно проверить с помощью утилиты *smartmontools* (http:// smartmontools.sourceforge.net); она, вероятно, присутствует и в репозитории вашего дистрибутива. К большинству дисков прилагаются диагностические инструменты, запускаемые с загрузочной дискеты, которая понадобится, если диск будет возвращаться по гарантии. Если в вашем компьютере нет дисковода, большинство диагностических программ можно запустить с загрузочного компакт-диска Ultimate Boot CD (www.ultimatebootcd.com). **НБ**

5 Стартовая лихорадка

В Несколько месяцев назад я заменил свою Mandriva на SUSE 10.2 с вашего LXFDVD91. У меня два внутренних жестких диска, оба разбиты на два раздела. Windows 2000 показывает их как **C:** и **D:** на диске 0 и **F:** и **G:** на диске 1. SUSE установлен на диске **F:**. Также у меня есть внешний диск, определяемый как диск **J:**. Когда я устанавливал первый Linux, то, к сожалению, подключил диск **J:**, и теперь, если я не включаю его при загрузке, курсор не перемещается в нижние пункты загрузочного меню.

Я хотел бы удалить Linux из системы полностью и начать заново с другим, более емким, вторым жестким диском. Однако ни в ваших учебниках, ни в справке программного обеспечения Linux как-то не просматривается средств, с помощью которых можно вернуться к моей простой Windows 2000 и HD0. Подозреваю, что если я сотру из Linux HD 1 (диск F в Windows), как мне сейчас хочется сделать, при включении компьютера вообще не будет появляться меню, так что я полностью сбит с толку.



Перестроить загрузочное меню из *YaST* в SUSE до крайности просто.

Надеюсь, что вы сообщите мне, как удалить Linux полностью, чтобы я мог вернуться туда, где был перед установкой Mandriva. Дав мне при загрузке выбор между Win2000 и XP без необходимости держать включенным внешний диск, вы получите мою признательность и вернете мне хороший сон.

Джон Бивен [John Beaven]

О Меню загрузки, вероятно, заработало бы, если бы вы немного подождали: похоже, что загрузчик *Grub* пытается читать отсутствующий диск, а для этой операции установлен тайм-аут. Чтобы это исправить, не нужно переустанавливать систему – просто измените настройки загрузчика. Это можно сделать из *YaST*. Загрузитесь при подключенном внешнем диске, затем отмонтируйте и отключите его или выключите питание диска. Запустите *YaST*, выберите System > Boot Loader > Boot Loader Installation »

Часто задаваемые вопросы

Есть ли вирусы для Linux?

Да, но они являются лишь доказательством возможности их написания. В настоящий момент вредоносных вирусов для этой ОС нет.

» Значит, беспокоиться не о чем?

Вы защищены, но не в безопасности. Нет вредоносных вирусов на момент написания статьи. Это не значит, что они не появятся в будущем или в момент чтения вами этих строк.

» Но ведь система прав доступа Linux не даст вирусу возможность причинить реальный вред?

Распространенное заблуждение! Конечно, если у вируса нет прав администратора, он не сможет удалить или переписать системные файлы, но спросите себя: что для вас ценнее – операционная система, которую можно переустановить менее чем за час, или ваши личные данные? Что вы скорее готовы потерять – пару библиотек или годовую финансовую отчетность перед подачей налоговой декларации? Вирус или троян может использоваться как спам-бот и без прав администратора, а это может заставить вашего интернет-провайдера приостановить ваш доступ к Сети.

» ОК, вы меня застращали. Как же защитить мою систему?

Существует несколько антивирусных программ для Linux. Самые популярные из них – *clamav* (www.clamav.net) и *F-Prot* (www.f-prot.com). Все они находят вирусы и для Windows, и для Linux. Если вы предпочитаете графический интерфейс – *KlamAV* (http://klamav.sourceforge.net) предоставляет интерфейс KDE для *clamav*.

» Какое мне дело до вирусов Windows на Linux-машине?

Если вы запускаете почтовый сервер, к которому подключаются клиенты Windows, имеет смысл избавиться от любых вирусов, прежде чем заняться чем бы то ни было, имеющим отношение к Windows. Большинство почтовых серверов можно настроить на фильтрацию почты через *clamav* для удаления или помещения в карантин подозреваемых на вирусы сообщений.

» В чем причина меньшей угрозы от Linux-вирусов?

Численность. Число пользователей Windows гораздо больше числа пользователей Linux. Кроме того [в Linux], используется больше различных программ. Подавляющее большинство пользователей Windows используют для работы с почтой *Outlook*; нацельте на него вирус, и жертвами падут большинство пользователей Интернета. Linux-пользователи могут выбирать из *KMail*, *Evolution*, *Thunderbird*, *Sylpheed*, *Mutt* и десятков других почтовых программ, доступных для Linux.

» Вы разве подразумеваете не вири?

Нет. Слова «вирусняки», «вири» могут быть подходящими множественным, но единственно правильное множественное от «вируса» – это «вирусы». Люди, пользующиеся общественным транспортом не ездят ведь на буусах!



***KlamAV* – KDE-интерфейс к антивирусу clamav, предоставляющий возможность ручной и автоматической проверки файлов и почтовых сообщений.**

» и выберите Propose New Configuration из всплывающего меню в правом нижнем углу окна. Имеющиеся диски (внешнего диска среди них теперь нет) будут просканированы, и создастся новое меню для установленных Windows и SUSE. Перейдите во вкладку Section Management, чтобы убедиться, что все сделано, как вы хотели, и нажмите кнопку Finish.

Если вы твердо решили удалить Linux с этих дисков, выберите пункт Restore MBR of Hard Disk из этого же меню, и код загрузчика заменится на существовавший до установки SUSE. Если это был загрузчик Windows, все нормально, но если вы перешли из Mandriva прямо на SUSE, при этом восстановится загрузочный код Mandriva, что вам совсем не нужно. Тогда вы должны загрузиться с компакт-диска Windows в режиме восстановления и запустить **fixmbr**, который ликвидирует загрузчик Linux и заменит его загрузчиком Windows.

В качестве альтернативы вы можете просто заменить второй диск (что, возможно, без вышеперечисленных действий нарушит порядок загрузки с жесткого диска), загрузиться прямо с установочного диска SUSE, инсталлировать его, и пусть он создаст новое загрузочное меню за вас, только не забудьте на сей раз отсоединить внешний диск. SUSE, как и все современные дистрибутивы Linux, вполне способен обнаруживать внешний диск при подключении его после установки операционной системы. **НБ**

6 Безопасный серфинг

Впрорываюсь в Linux уже в течение нескольких месяцев. Я пробовал загружать дистрибутивы, но столкнулся с большим количеством технического жаргона. А вот загрузил Ubuntu 7.04, и полюбил его. В моей голове до сих пор не укладывается факт, что это все бесплатно, так же, как и программное обеспечение, поставляемое с дистрибутивом, но я уверен, что привыкну к этому.

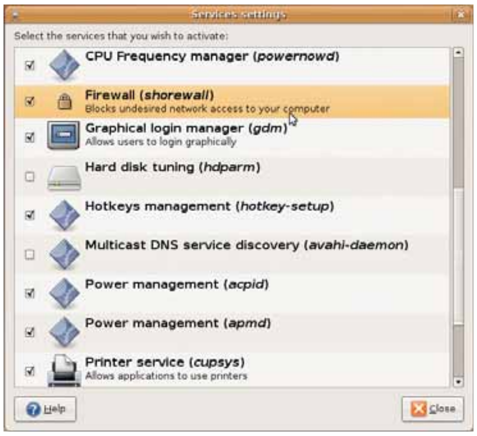
Так как я новичок, то должен не раз убедиться, что мои действия безопасны, и я не открываю мой ПК для внешних хакеров. Что я должен предпринять для установки брандмауэра и антивируса?

У меня установлена Ubuntu 7.04 в двойной загрузке с Windows XP Home. На XP у меня *F-Secure 2007* – комбинация брандмауэра и антивируса. Я подключен к Интернету с использованием внешнего модема – маршрутизатора через сетевой кабель.

Стив Холл [Steve Hall]

ОВирусы, в общем-то, не проблема в Linux, но все равно лучше предохраниться. Самая популярная антивирусная программа под Linux – *ClamAV* (www.clamav.net), она включена в Ubuntu, и ее можно установить через менеджер пакетов *Synaptic*. *ClamAV* хорошо распознает вирусы и Windows, и Linux, поэтому, скомбинировав его с модулями расширений для самых популярных почтовых программ, Вы можете быть уверены, что никакие вредные вложения не достигнут Windows через вашу почту в Linux.

В Linux брандмауэр трактуется иначе, чем в Windows. Отсутствие шпионских программ и практическая невозможность встраивания их в программы с открытым исходным кодом означают, что основная задача – это предотвращение вторжений. В Linux ПО фильтрации сети встроено в ядро, а различные брандмауэры дают более или менее простые способы создания, проверки и применения правил фильтрации. Есть несколько заслуживающих рассмотрения пакетов в репозитории Ubuntu, в том числе: *Firewall Builder* (www.fwbuilder.org), *Guarddog* (www.simonzone.com/software/guarddog) и *Shoreline Firewall* (www.shorewall.net). Первый – это Qt-программа, хорошо вписывающаяся в рабочий стол по умолчанию KDE; *Guarddog* является приложением *GTK+*. Они предлагают аналогичные функции, но разные подходы. *Shoreline Firewall* – программа на основе скриптов, безусловно, трудная при первой установке, но обеспечивающая большую гибкость. Любая из них способна защитить вашу систему, поэтому стоит установить их и посмотреть, какая вам больше всего понравится.



› **Выключение ненужных служб повышает безопасность, но вот брандмауэр как раз выключать не стоит!**

Вы также должны уменьшить шансы «захватчиков» вообще добраться до брандмауэра. Ваш маршрутизатор является первой линией обороны, так что отключите те порты, которые вы не используете. Отключите также все неиспользуемые службы Ubuntu, в окне System > Services, но будьте осторожны: некоторые из них необходимы для нормальной работы компьютера. Если вы не уверены, отключайте сервисы по одному и смотрите, что у вас получилось, чтобы в случае проблем можно было включить их обратно.

Хотя Linux по своей сути безопаснее, чем Windows, не нужно слепо на него полагаться – Linux-программы также могут иметь уязвимости. Они, как правило, оперативно исправляются, так что регулярно обновляйте свою систему. Четыре шага: блокировки на маршрутизаторе, отключение ненужных служб, запуск брандмауэра и постоянное обновление ПО – означают, что вы можете спокойно пользоваться Интернетом, не подвергаясь опасности. **НБ**

7 Подкачнули Vista

ВПытался создать раздел подкачки на старом MP3-плейере, а случайно сделал файл подкачки на разделе с Vista. Я этот файл не подключал, а чтобы его сделать, использовал команду **mkswap**. Vista все еще работает, хотя приходится загружаться через восстановительный раздел, и я догадываюсь, что задел только начало диска. Можно ли «откатить» команду **mkswap**, чтобы исправить раздел с Vista?

Я проверил диск *GParted*, и он показал его как диск подкачки. *Fdisk* показывает раздел как NTFS, как и должно быть, но в загрузочном заголовке нет *. Означает ли это, что я просто могу восстановить загрузочную информацию, и все заработает?

celticbhoy, с форумов LXF

ОРаз Vista все еще работает, значит, раздел в порядке. Похоже, вы изменили тип раздела – вероятно, на Linux Swap – и очистили флаг загрузки. В итоге Windows не может распознать раздел, и начальный загрузчик думает, что загрузиться отсюда нельзя. Используя любой редактор разделов, вы можете выбрать для него тип NTFS (07) и установить флаг загрузки. Я убедился, что с этим легко справляется *cfdisk*; вдобавок он есть на любом из известных мне Live CD. Загрузитесь с Live-диска, откройте терминал от root и запустите *cfdisk*:

```
cfdisk with:
cfdisk /dev/hda
```

Сделав это, выберите раздел, нажмите t, чтобы задать тип, и выберите NTFS из списка вариантов; затем нажмите b и сделайте его загрузочным. В заключение нажмите W (заглавное) для записи изменений на диск. Можно сделать это и при помощи редактора с графическим интерфейсом, вроде *GParted* или *qtparted*, но я нашел, что *cfdisk* быстрее. Вам даже не нужно ждать загрузки рабочего стола, если ваш любимый Live CD имеет опцию загрузки прямо в командную строку (пользователи Knoppix, например, могут набрать **knoppix 2** в строке загрузчика). **НБ**

8 USBить Samba

ВЯ установил небольшой сервер под Debian Etch, в основном как файл-сервер, но потом приспособил его и для некоторых задач web. У меня есть жесткий диск с USB, и я хочу использовать его как устройство хранения через *Samba*. Моя проблема заключается в том, что как я его ни подключаю, он всегда монтируется с правами администратора. Если я устанавливаю точке монтирования права **0777, user=guest** и **group=users** и монтирую его как обычный пользователь, права остаются прежними, но пользователь и группа возвращаются на root. Поэтому я ничего не могу записать на диск. Если я монтирую его от пользователя root, то у меня нет проблем при локальном доступе, но в любой ситуации *Samba* не позволяет мне на него писать.

Кто-то предположил, что это, возможно, из-за *udev* и что мне нужно было перешерстить его настройки, чтобы изменить разрешения при монтировании. А я не разбираюсь в *udev* и не знаю, с чего начать. Диск определяется как **sda** с разделами **sda1** и **sda2**.

Энди [Andy]

О*Udev* занимается только созданием узла устройства (**/dev/sda1** или любого другого), а не монтированием, так что вряд ли может быть виноват. Вполне возможно, что *udev* создает узел с ограничительными правами, но это лишь остановит монтирование устройства пользователями (не root) и не скажется на уже смонтированных файловых системах.

Опция монтирования user не требует его имени – она просто позволяет любому пользователю монтировать файловую систему, и не влияет на ее разрешения. Решение вашей проблемы зависит от типа используемой файловой системы. Если это одна из файловых систем Linux, поддерживающая права пользователя, достаточно установить владельца и права доступа для точки монтирования, но только после того, как файловая система была смонтирована, в противном случае вы установите параметры для точки монтирования, а

Вопрос-победитель (английская версия)

Небезопасный FTP

В У меня внешний сервер, работающий как FTP для сотрудников компании, а также как анонимный FTP-сервер для наших клиентов. Еще недавно это был милый маленький сервер, но мы обнаружили, что им злоупотребляют сторонние пользователи помимо наших клиентов, съедая пропускную способность нашей сети.

Проблем не бывает, если сервер запускается после работы, не влияя на работу сотрудников, которые используют эту же сеть в течение дня. Поэтому я через задание cron останавливаю его утром и запускаю после закрытия офиса. Проблемы с пропускной способностью сети исчезли, но возникла другая проблема. Сотрудникам, которым необходимо обновить файлы на FTP-сервере, приходится делать это после работы.

Мне нужно, чтобы сервер запускался для местных сотрудников, а отнюдь не со стороны. У меня есть кое-какие мысли, но они связаны либо с изменением файлов **hosts.allow/hosts.deny**, либо с использованием трюков с *xinetd*. Хорошего решения я не вижу.

Сервер работает на CentOS 4.5, используя *vsftp*, запущенный как автономный демон. Машина имеет только одну сетевую карту и IP-адрес и видна только через этот адрес.

Пит Дарст [Pete Durst]

О То, чего вы хотите, можно добиться, сделав доступным публичный сервер только вне рабочего времени, но это не идеальное решение. Разумно предположить, что те, кто злоупотребляют сервером, не всегда выкладывают на нем юридически чистые материалы, что может привести к судебному иску против вас или потере соединения с Интернетом. Напомним, что это ваш сервер, и вы несете ответственность за его содержание.

Предоставление анонимных закачиваний и скачиваний ведет к потенциальным неприятностям. Если все-таки это необходимо делать, держите области закачки отдельно от области скачивания, чтобы люди не могли скачать материалы, присланные анонимно: они должны быть связаны с чьей-то учетной записью. Лучшее решение – отключить анонимную закачку и предоставить Вашим клиентам личные учетные записи FTP.

Если Вы действительно хотите продолжать предоставлять неограниченный анонимный доступ в нерабочее время, воспользуйтесь файлами **hosts.allow** и **hosts.deny** в **/etc**, поместив

```
tcp_wrappers=YES
```

в **/etc/vsftpd.conf**. Затем, для обеспечения доступа из локальной сети, добавьте строку в **/etc/hosts.allow**:

```
vsftpd: 192.168.1.
```

Заметьте: адрес пишется до последней точки, чтобы соответствовать целой подсети. Измените его на адрес своей сети. Теперь добавте такие две строки в **/etc/cron.d/vsftpd**:

```
0 18 * * 1-5 root sed -i '/^vsftpd/d' /etc/hosts.deny
0 8 * * 1-5 root echo "vsftpd: ALL" >>/etc/hosts.deny
```

и заставьте *cron* перегрузиться:

```
killall -HUP cron
```

Это приведет к запрету изменения **hosts.deny** для запрета всех адресов, кроме указанных в **hosts.allow**, между 08.00 и 18.00 с понедельника по пятницу и удаления блокировки в любое другое время. **МС**

не для смонтированной файловой системы.

В файловых системах Windows, в частности, FAT32, вы можете добавить параметр **umask=002** в **/etc/fstab**, чтобы сделать все файлы доступными для чтения и записи пользователю и группе. Затем используйте опции **uid** и **gid** для установки владельца всех файлов файловой системы. Здесь вы можете использовать числовые значения, или имена пользователя и группы, например:

```
/dev/sda1 /mnt/somewhere vfat umask=002,uid=guest,gid=users 0 0
```

НБ

9 Почтовая петля

В При настройке связки *Postfix+SpamAssassin+ClamAV* возникла сложность с перенаправлением спама на внутренний почтовый адрес (это необходимое условие, поскольку я хочу изучать его и вылавливать ошибочно отброшенные сообщения). Если указать в **procmailrc** **! fail@gts.example.ru**, то почтовую систему зацикливает от одного спам-сообщения, а журнал почтового сервера растет прямо на глазах. Если же написать **procmailrc** вида

```
MAILDIR=/home/fail

:0fw
| /usr/bin/spamc --username=fail

:0
* ^Subject:.*(SPAM|spam)
$MAILDIR/spam/
```

то все работает почти как надо, но спам-сообщение попадает в каталог **spam** с правами пользователя-респондента, а не **fail**, который должен его читать. Подскажите, где ошибка?

Артур

О Проблема в том, что когда вы перенаправляете письмо пользователю **fail@gts.example.ru**, оно снова попадает на обработку MTA. *Procmail* опять видит строчку SPAM в поле Subject, передает письмо пользователю **fail** и так до бесконечности, а точнее – до предела полей Received в заголовке. Рецепт простой – настроить **procmailrc** так, чтобы все письма для пользователя **fail** уже не обрабатывались. Например, в начало **procmailrc**, до разбора строчки *** ^Subject:.*(SPAM|spam)** можно вставить что-то вроде

```
:0
* ^TO_fail@gts.example.ru
/home/fail/.maildir/
```

Должно сработать. **ВИ**



› *Samba* имеет множество настроек, но первоочередная задача – гарантировать возможность чтения каталогов, к которым предоставляется доступ.

Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Большой вопрос

На этом раздел заканчивается, но наши эксперты не расходятся по домам до тех пор, пока не дойдут до сути общей проблемы.

10 Установить сервер

В Я устанавливаю web/почтовый сервер. Что бы вы порекомендовали новичку как дружественный и безопасный Linux-дистрибутив и ПО для web и почтового сервера с простым интерфейсом для обмена файлами с Windows? Желательно что-нибудь с подробным руководством пользователя, он будет использоваться в небольшом офисе.

Алекс [Alex]

О О боже, мне ясно одно: что бы я ни порекомендовал, это вызовет всплеск несогласия пользователей других дистрибутивов. Популярные дистрибутивы годятся и для настольного использования, и для рабочих станций или серверов, но есть ряд мелких дистрибутивов, специально созданных для задач, подобных вашим. Один из таких дистрибутивов – ClarkConnect (www.clarkconnect.com). ClarkConnect больше известен как интернет-шлюз и содержит средства подключения к сети Интернет с соответствующими фильтрами контента и контролем доступа, но также может использоваться в качестве сервера интранет.

Исходя из того, что Вы новичок и будете использовать сервер в коммерческих целях, настоятельно рекомендую Вам рассмотреть возможность использования платной версии, хотя для пробы можно установить бесплатную версию. ClarkConnect предлагается в версиях Community, Office и Enterprise. Community Edition абсолютно бесплатна, а две другие имеют 30-дневный пробный период. В платных версиях предоставляются дополнительные функции и, самое главное, поддержка.

ClarkConnect требует подключения к серверу клавиатуры и монитора только на время установки и базовой начальной настройки, после чего все делается через web-интерфейс. Администрирование выполняется через защищенное SSL-соединение на нестандартном порту, поэтому, зная IP-адрес сервера ClarkConnect, который вы увидите после входа в систему как root, можно использовать любой web-браузер в сети для подключения к http://ip-адрес:81.

Установка Linux дает безопасность в том случае, если вы ее обеспечите. Использование сервер-ориентированных дистрибутивов с минимальным числом пакетов помогает в этом, но только если постоянно обновляется ПО, используется меню Software Install в административном интерфейсе, и у вас установлен брандмауэр. Брандмауэр уже включен в установки и настраивается из браузера. Даже если вы не используете сервер в качестве интернет-шлюза, разумно защищать сервисы и данные в системе отдельным брандмауэром, в дополнение к имеющемуся на вашем шлюзе или маршрутизаторе.

Для решения вашей задачи пригодны и дистрибутивы общего назначения, если выбрать установку серверных вместо настольных пакетов. Большинство дистрибутивов включают прекрасную программу Webmin, позволяющую управлять серверами из браузера, но для отдельного сервера, особенно если ваш опыт ограничен, специализированный дистрибутив, вероятно, наилучший выбор.

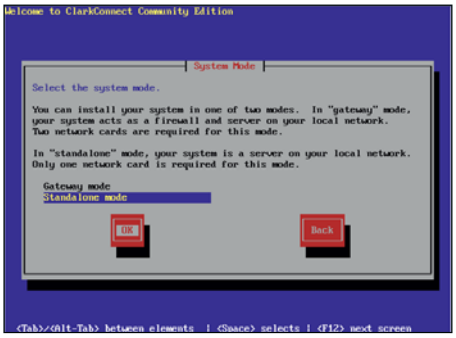
Хотя ClarkConnect не комплектуется печатными руководствами, множество документации есть на сайте: и подробное руководство пользователя, и различные How To. Существуют также форумы для взаимопомощи. **НБ** LXF

Шаг за шагом: Знакомимся с ClarkConnect



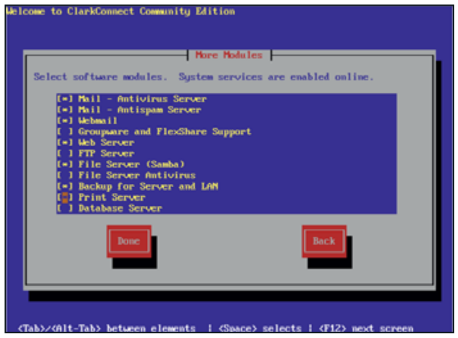
1 Запускаем установщик

После скачивания ISO-образа с сайта ClarkConnect и записи его на CD, загрузитесь и начните процесс установки.



2 Шлюз или внутренний?

ClarkConnect может работать как интернет-шлюз или отдельный сервер в вашей сети. В обоих случаях предлагается выбор серверов.



3 Выбор ПО

Выберите нужные типы серверов. Выбирайте только те, что нужны вам сейчас; если понадобится, другие вы сможете установить через web-интерфейс.



4 Первый запуск

Вот это вы увидите, войдя в ClarkConnect на своей машине, хотя легче управлять им дистанционно через браузер.



5 Простая настройка

Сервера и все прочее настраивается дистанционно с помощью web-интерфейса, без подключения к серверу клавиатуры или монитора.



6 Подробная документация

На сайте ClarkConnect достаточно информации и справочных материалов. Большинство разделов web-интерфейса имеют ссылки на соответствующие документы.

Лучшие новинки открытого ПО на планете

LXF HotPicks



Пол Хадсон
Отслеживая все последние разработки в приложениях с открытым кодом, наш замредактора завел пару своих Hot Picks: он ведет проекты *Fuzz Engine* и *Brain Party на SourceForge*.

» Wiki On A Stick » Zero Install » Herrie » JSCoverage » Slingshot
» Thunder & Lightning » Segatex » Bitswash

Wiki в вашем браузере

Wiki on a Stick

LINUX FORMAT Hottest Pick

Версия v0.9.2B **Сайт** http://sourceforge.net/projects/stickwiki

LINUX FORMAT На DVD

История можно разделить на две эпохи: до наклеек Post-it и после. Но даже маленькие желтые листочки 3M не могут справиться с горами информации, которые нам приходится перекапывать сегодня; поэтому мы представляем вам новое мощное средство: *Wiki* on a *Stick* (*WOAS*).

Всем известно, как работают *Wiki*-сайты: мгновенное редактирование страниц, упрощенный синтаксис форматирования, перекрестные ссылки в статьях, и прочее. Эту идею *WOAS* реализует с помощью одного-единственного HTML-файла. Вы загружаете его в любой браузер на ваш выбор и получаете полностью сконфигурированный готовый *Wiki*. Все изменения сохраняются в этот файл, так что вам остается просто скопировать его на USB-брелок вместе с Portable *Firefox*, и ваши данные везде будут с вами.

Магии подобно

Согласно третьему правилу предсказаний Артура К. Кларка, любая достаточно передовая технология неотличима от магии, и именно так и выглядит *WOAS* – с трудом верится, что такая функциональность уместилась на простой web-странице. Причем именно так *WOAS* и работает: каждая страница закодирована внутри HTML-файла и обвешана множеством сценариев JavaScript, помогающим магии свершиться. Одно это само по себе очень впечатляет, но *WOAS* творит еще и такие чудеса, которые вам и не снились: поиск, шифрование (!), пространства имен, печать и даже вставка изображений – и все это как стандарт.

> «Поиск, шифрование, пространства имен, печать и даже вставка изображений!»

Вы даже можете добавлять скрипты JavaScript к вашим страницам, чтобы получить новые функции – как пример, включена игра, показывающая, как это делать. Вы можете спросить: почему *WOAS* лучше, чем, допустим, Tomboy; но *WOAS* имеет ряд ключевых преимуществ. Она основана на браузерной технологии, и вы можете работать со своими данными в любом поддерживаемом браузере, не будучи заложником Gnome/*GTK*-интерфейса Tomboy. Кроме того, данные хранятся в JavaScript, а значит, при желании очень легко написать свой код, чтобы побаловаться с ними. Наша любимая функция – AES-шифрование: нажмите на символ замка в правом верхнем углу каждой страницы, чтобы зашифровать ее паролем; *WOAS* даже показывает степень стойкости пароля, когда вы его набираете.

Версия *WOAS* еще не достигла номера 1.0, то есть программа находится в активной разработке. Даниэле К. [Daniele C.], руководитель проекта *WOAS*, сообщает: «Планируется версия v0.9.4B, где добавятся другие функции – например, сохранение с накоплением (чтобы свести к минимуму число обращений к диску) и автосохранение через заданный промежуток времени». Итак, *WOAS* еще похорошеет; но я особо предвкушаю реализацию другой идеи Даниэле – добавления импорта/экспорта XML: это здорово упростит доступ к данным.

Шаг за шагом: Осваиваем Wiki



» Опустошим установки
WOAS имеет содержание по умолчанию, но если вы пользовались wiki до этого, можете удалить его, просто нажав на ссылку «Очистить установки по умолчанию».



» Добавим содержание
Нажмите на карандаш (вверху справа), чтобы добавить текст на первой странице. Разместите текст внутри [[и]], чтобы сделать ссылку, например [[Hello]]. Затем щелкните на значке диска для сохранения.



» Заполним ссылки
Все созданные вами ссылки будут показаны красным цветом, так как они не имеют содержания по умолчанию. Нажмите на любую из них, чтобы создать новую страницу.

Децентрализованный менеджер пакетов

Zero Install

Версия 0.29 **Сайт** www.0install.net



Менеджеры пакетов в большинстве своем достаточно «умны», но всегда найдется какая-нибудь программа, которую вы не сможете добыть для своего дистрибутива. Можно бы скомпилировать из исходников, да ведь намучаешься, отслеживая множество зависимостей вручную! Так были придуманы пакеты *Autopackage*: предварительно скомпилированное, простое в установке программное обеспечение, работающее под любым дистрибутивом Linux.

Но и тут имеются проблемы, и тогда на помощь приходит *Zero Install Injector* (*ZII*). К примеру, *Autopackage* дает большие возможности: обычные (не супер-) пользователи могут установить свое собственное ПО (поместив его в свой домашний каталог), но что произойдет, если программу попытаются установить два, три или более пользователей? Ответ: ПО многократно скопируется в разные места на жестком диске, а это напрасный расход памяти.

ZII решает эту проблему путем превращения программы в URL-адреса: при каждом запуске приложения компьютер загружает XML-файл из Интернета и проверяет его на наличие обновлений относительно того, что вы установили. Если ПО установлено вами или другим пользователем, оно запускается как обычная программа, добавленная через менеджер пакетов. А если ПО не установлено или доступен обновленный вариант, *ZII* автоматически загружает скомпилированный пакет и устанавливает его, а затем запускает, как если бы программа была инсталлирована заранее. Если Интернет у вас не подключен, *ZII* просто запустит программу, не пытаясь установить соединение.

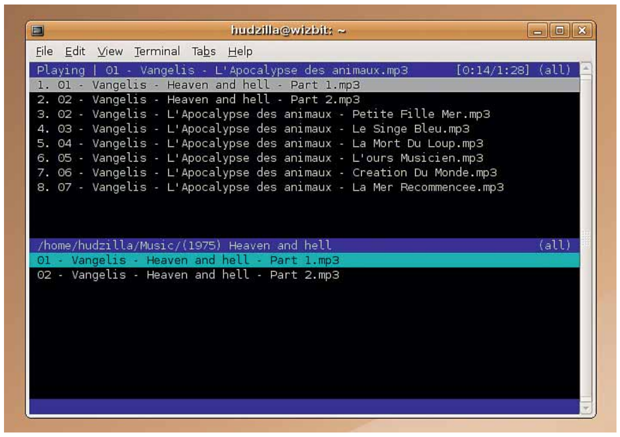
Вы можете подумать, что в Linux и так уже хватает менеджеров пакетов, но *ZII* превосходит их всех, вбирая лучшие черты и объединяя их в одну функцию. APT заставляет вас использовать файлы **sources.list**, *Autopackage* не позволяет иметь несколько версий программы на одном компьютере, *Klik* не поддерживает цифровые подписи для аутентификации программного обеспечения, а *ZII* обладает всеми этими функциями. «Запуск» URL может показаться странным, но это так невероятно легко делается, что потом вы будете дивиться, как мы жили без него. Как говорит создатель *ZII*, «В *Autopackage* речь в основном идет о подготовке интегрированного дистрибутива, перемещаемых двоичных файлов, с такими инструментами, как *apbuild*, *binreloc* и *relaytool*. *Zero Install* не охватывает все эти моменты (мы просто предлагаем использовать инструменты *Autopackage*), но предполагается, что у вас уже есть соответствующий двоичный архив, готовый для распространения».

› Пакеты *ZII* часто строятся поверх пакетов *Autopackage*, поэтому с достаточно большим количеством программного обеспечения можно работать легко.

Консольный аудиоплейер

Herrie

Версия 1.8.1 **Сайт** www.herrie.info



Все еще пользуетесь *mpg123* для прослушивания музыки? Позвольте **LXF** представить вам кое-что поудобнее, более подходящее для проигрывания плей-листов на вечеринках. *Herrie* – это медиа-плейер, работающий из командной строки, он позволяет легко находить и сортировать музыку в соответствии с вашими вкусами. Его компиляция не очень-то легка, поскольку он проигрывает всевозможные звуковые форматы, которым требуются свои библиотеки. Если вы работаете в Ubuntu, то знайте, что мы добились лучших результатов, сперва установив зависимые пакеты: *curl*, *libcurl3-dev*, *libmodplug-dev*, *libncurses5-dev*, *libncursesw5-dev*, *libvorbis-dev*, *libsndfiledev*, *libmad0-dev*, *libid3tag0-dev* и *libasound2-dev*. У других дистрибутивов будут похожие имена пакетов, а может, даже библиотека XSPF – в Ubuntu ее нет. После установки библиотек мы выполнили configure с флагом "no_xspf", потому что эта библиотека не была установлена, а потом, как обычно, выполнили *make* и *make install* от имени суперпользователя, и *Herrie* был готов к действию.

По умолчанию *Herrie* запускается с такими установками, что песни после воспроизведения удаляются из плей-листа, во избежание повторов. Если вас это не устраивает, просто запустите *Herrie* с параметром -x, чтобы перевести его в режим *XMMS*, и вы сможете использовать клавиши b и z для перехода к последующему и к предыдущему треку. (Регистр имеет значение, вы сразу это заметите.)

Простота эксплуатации

Основной интерфейс *Herrie* разделен на две части: вверху вы видите композиции в плей-листе, а внизу – файловую систему, управлять которой можно с помощью курсора. Чтобы добавить песню или каталог в плей-лист, просто нажмите a.

Как любое порядочное приложение, работающее из командной строки, *Herrie* располагает большим количеством клавиатурных комбинаций для выполнения общих задач; некоторые из них отличаются в зависимости от регистра. Например, r устанавливает режим повтора, тогда как R добавляет функцию рандомизации плей-листа (просто нажмите "y" – «да» – по запросу). Кроме того, d удаляет песню из списка воспроизведения, а D удаляет все песни из плей-листа. Особо понравились нам функция поиска (наберите /, затем критерии поиска) и быстрая перемотка вперед-назад (> и < соответственно), но, несомненно, лучшее свойство *Herrie* – то, что он почти не тратит ресурсы центрального процессора и оперативной памяти. Вы не найдете программы быстрее и легче этой.

> «В Herrie, как в порядочном приложении, работающем из командной строки, много горячих клавиш.»

› Приложения, работающие из командной строки, часто имеют сложный интерфейс пользователя, но *Herrie* стремится к простоте и удобству – и преуспевает в этом.

Проверка Javascript

JSCoverage

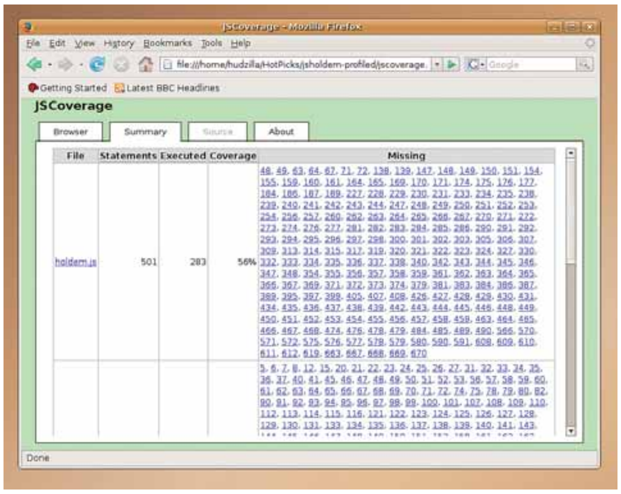
Версия 0.2 **Сайт** www.siliconforks.com/jscoverage

Практика использования *Wiki* on a *Stick* показывает, что большие приложения JavaScript привлекают все большее внимание. Фактически все кнопки «обновить-не-перегружая» на самых популярных сайтах Web 2.0 работают на JavaScript, а это означает, что крутая выделенная линия высокоскоростного Интернета, которой вы недавно обзавелись, уже насыщена тысячами строк кода JavaScript. Чтобы пресечь раздувание объемов JS, разработчики используют инструмент *JSCoverage*: он следит за выполнением JavaScript-кода, а затем сообщает, какие ветви кода используются, так что вы точно знаете, что можно подрезать. Он также полезен для выявления случайных логических ошибок, где условное утверждение выполняется всегда (или никогда), из-за путаницы в коде.

Как это работает

JSCoverage – вполне стандартная двоичная программа (просто используйте **./configure**, **make** и **make install** для установки), но она работает за счет вставки в JavaScript специального профилирующего кода, который затем отслеживается. Тестируя *JSCoverage* перед включением его в HotPicks этого месяца, мы использовали *JSHoldem* (забавный техасский вариант покера, см. *HotPicks*, **LXF91**). Для запуска программы применяется команда **jscoverage input_dir output_dir**, например, **jscoverage jsholdem jsholdem-profiled**. Все файлы из каталога **jsholdem** скопируются в **jsholdem-profiled** (создаваемый автоматически), а затем во все найденные сценарии JavaScript будет добавлен специальный код. Естественно, следует использовать два различных каталога, потому что исходный код JavaScript модифицируется физически!

> «Сделайте отдельный каталог для результата, так как исходный код JavaScript будет модифицирован.»



› Все неиспользуемые строки в вашем JavaScript-коде отображаются как ссылки – нажмите на одну из них, и попадете прямо в нужную точку исходного кода.

Обработав код, откройте ваш браузер и запустите **file:///path/to/JSHoldem-profiled/scoverage.html**; вы увидите графический интерфейс *JSCoverage*. Введите путь к вашему основному HTML-файлу в поле URL, нажмите кнопку Go, и он загрузиться в плавающее окно. Повзаимодействуйте с JavaScript, сколько требуется, затем нажмите Summary, чтобы узнать, какие именно строки были приведены в действие.

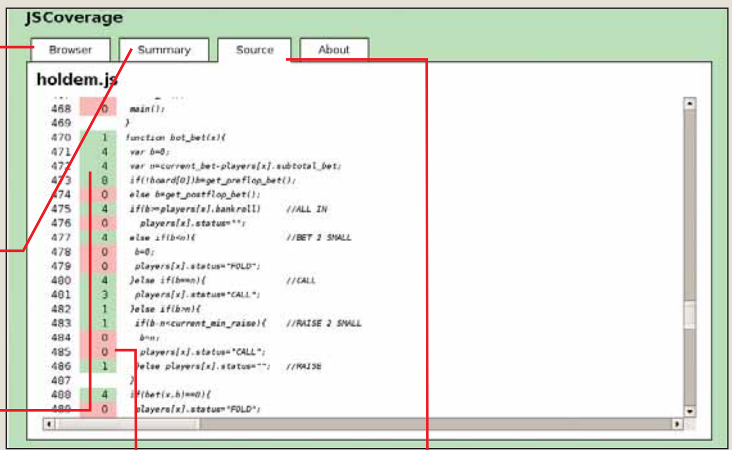
Анализ кода

Во вкладке Summary все неиспользованные строки кода будут показаны под заголовком Missing как ссылки, но для быстрого обзора смотрите Coverage percentage для каждого файла. Нажав на одну из ссылок под Missing, вы перейдете к просмотру исходного кода, где слева приписаны номера строк, а также количество запусков каждой строки.

Этот подсчет покажет вам, какие строки вообще не использовались (и можно задуматься, не исключить ли их, или не нуждается ли ваш код в срочной переделке, чтобы они все-таки запускались в нужный момент), и вы можете увидеть, какие ветви используются чаще всего, а значит, являются первыми кандидатами на оптимизацию. Конечно, между количеством переходов к ветви и временем, требуемым для ее исполнения, прямой связи нет, но это все же некая подсказка, с чего начать.

Если вы серьезно занимаетесь JavaScript и регулярно создаете скрипты на сотню-другую строк, *JSCoverage* – для вас: это отличный инструмент углубленного анализа, способный революционизировать отладку приложений.

Работаем с JSCoverage



Браузер
Это вид по умолчанию. Здесь можно запускать URL и взаимодействовать с вашим приложением.

Сводка
Обзор результатов, показывающий, какая часть каждого скрипта была выполнена.

Хорошо
Зеленая цифра 8 означает, что эта строка была запущена 8 раз, и, возможно, ее стоит оптимизировать.

Плохо
Красный 0 означает, что этот код вообще не запускался – нехороший признак для приложения.

Исходники
Здесь можно видеть исходный код с показанным количеством обращений к строкам.

3D-имитатор

Thunder & Lightning

Версия 070710 **Сайт** http://tnlgame.net



Игра *Carrier Command* была хорошим сочетанием воздушных, морских и наземных боевых сражений; подобных мы до сей поры не встречали. *Thunder & Lightning* (*TnL*), быстро развивающийся 3D-имитатор с открытым исходным кодом, стремится перенять опыт Carrier, но с обновленной графикой и более агрессивным «искусственным интеллектом». Все только начинается, и вы иногда можете даже позлиться, со свистом проносясь по небу, но уже на данном этапе разработки есть удачные достижения: установка, например, целиком идет через *Autopackage*, без всякой мороки.

Коль скоро ваша видеокарта поддерживает 3D, единственная потенциальная проблема – это *OpenAL*: если аудиокарта не совместима с ним, для отключения звука придется потрудиться. После некоторых мучений, мы пришли к достаточно простому решению: создайте файл **.openalrc** в вашем домашнем каталоге и вставьте следующее:

```
(define devices '(oss))
```

Установив Автопакет, наберите **tnl**, чтобы запустить игру. Обучающий режим поможет взлететь, набрать высоту и приземлиться.

Думаете, просто? Мы сумели правильно приземлиться только с седьмой попытки, да и то благодаря случайному везению! Жмите l, чтобы пропустить обучающий режим и перейти прямо к игре; затем a, чтобы перейти на ручной контроль, и 0 для перехода к максимальной скорости – и вы в полете! Управление полетом состоит из движений мышью (направление), стрельбы (левая кнопка мыши – для стрельбы, правая – для смены оружия) и клавиатуры, но все, что вам действительно понадобится – это клавиши переключения скорости (1–0), посадочное шасси (L) и швартовочный крюк (K). Миссии очень сложны, но и просто полетать вокруг и пострелять по танкам и прочим целям тоже забавно. Создатель игры Йонас Эшенбург [Jonas Eschenburg] говорит: «Самая востребованная функция – многопользовательский режим. К сожалению, мне пока не хватает ресурсов и знаний, чтобы его запустить. Помощь имеющих опыт в этой области пришлась бы очень кстати!»

› Хотите беспощадной пальбы? TnL осуществит вашу мечту, и довольно реально!

> «Миссии очень сложны, но и просто полетать – весьма забавно.»

Стрельба в космосе

Slingshot

Версия 0.8p **Сайт** www.slingshot-game.org



Помните *Gorrillas*, старую MS DOS-игрушку от IBM? Две огромных гориллы, стоя на башнях, кидались друг в друга разрывными бананами, пока одна из них не погибала? Вам нужно было выбрать силу и угол для броска, в надежде, что ваши бананы попадут в соперника или, по крайней мере, взорвут его укрытие. Играть было довольно весело; *Slingshot* добавляет новый блеск этой идее, делая ее покруче, чем раньше: *Gorrillas* вернулась, и на сей раз – с гравитацией!

Стрельба бананами

Игра *Slingshot* разворачивается в космосе, где два межпланетных корабля с мощными лазерными пушками жаждут уничтожить друг друга. Управление: стрелки курсора для наведения и придания ударной силы, и пробел для залпа. Ваша миссия: засекать все, что шевелится, и палить по гадам, прежде чем они смогут выпалить в вас. Однако на траектории выстрела находятся планеты разных размеров, и их гравитация воздействует на луч, так что целиться нужно очень тщательно. Стрелка «вверх» добавляет мощи выстрелу; но, задав эту мощь с перебором, вы рискуете пробить свою мишень насквозь, и луч вашего лазера вернется по кругу на вас самих.

Slingshot – не одно лишь уничтожение друзей и коллег в звездной войне: очки также выставляются за стиль и за скорость. То есть, к 1500 очкам за поражение цели может прибавиться величина еще до 500 очков, если вы попали с первых трех раз. Зато вычитается «штраф», зависящий от мощи выстрела, чтобы поощрить игроков использовать более слабые выстрелы, искусно эксплуатируя силы гравитации, с целью набрать максимум очков.

Понятно, что *Slingshot* – игра не для мыслителей, но она доставляет массу удовольствия от серий коротких очередей, особенно если у вас живой противник. В версии 1.0 обещан многопользовательский режим, и это было бы здорово. Но также предполагается переписать код на C++, что уже кажется малость сомнительным!

› Коллега LXF – журнал *PC Format* – наслаждается игрой с открытым исходным кодом и элементом гравитационного безумия.

> «Эксплуатировать силы гравитации, с целью набрать максимум очков.»

Инструменты SELinux

Segatex

Версия 3.04 **Сайт** www.sourceforge.net/projects/segatex

Всем известно, что безопасность – дело хорошее, и программное обеспечение, улучшающее безопасность без особых для вас неудобств, является особенно ценным. Для пользователей Fedora, SELinux – один из программных продуктов переднего края борьбы со злоумышленниками, но также и последнее слово в «непознаваемости» пользовательских интерфейсов: она «просто работает», и горе тем, кто пытается понять, как. Однако если уж вы системный администратор, то обязаны понимать принципы ее работы. Представляем вам *Segatex*: это крошечная программка, призванная облегчить редактирование политик SELinux и объединяющая целый ряд других инструментов SELinux.

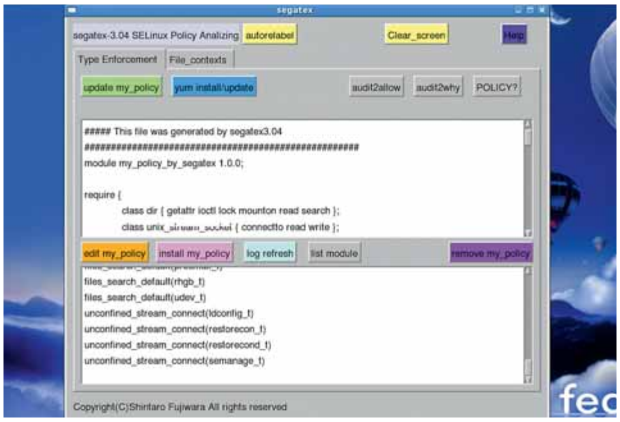
Легче, но не легко

Чтобы сделать SELinux простым в использовании, требуется умеренной степени чудо, но *Segatex*, по крайней мере, делает его чуть более управляемым. В Fedora, единственном, помимо RHEL, дистрибутиве, полезном для *Segatex*, нужно установить библиотеки разработчика *Qt* и *Boost*. После этого вы можете просто использовать RPM для установки *Segatex*. При желании попробовать его на других дистрибутивах (SELinux можно использовать под Debian, но «из коробки» он для этого не сконфигурирован), нужно компилировать исходный код. Я запускал сборку на нескольких дистрибутивах и каждый раз обнаруживал ошибки компиляции, но их, как правило, было легко исправить в *Makefile*, добавив это в конец строки, отмеченной **LIBS**:

```
-lselinux -lboost_regex
```

Запустив *Segatex* (вы должны быть суперпользователем), приступайте к редактированию файла **my_policy**, по нажатию оранжевой кнопки. Изменения можно добавлять в нижней части панели, а когда вы закончите, жмите на розовую кнопку, чтобы изменения вошли в силу. В закладке File Context можно просмотреть и изменить регулярные выражения, которые SELinux использует для привязки контекста к файлам, и сразу же увидеть результаты, нажав на голубую **ls- laZ** кнопку.

> «Segatex не отменяет необходимости понимать работу SELinux.»



› ***GUI Segatex* использует цвет, чтобы было легче запомнить, какие кнопки безопасно нажимать. Будьте осторожны!**

Имейте в виду, что SELinux – система очень сложная, и даже *Segatex* не отменяет необходимости понимать принципы ее работы. Более того, любые сделанные вами изменения применяются к системе в целом, так что зря по кнопкам не стукайте! И последнее: по нажатию на желтую кнопку Autorelabel все ваши файлы при перезагрузке будут маркироваться заново, а это процесс длительный; лучше оставить эту кнопку в покое.

Клиент BitTorent

Bitswash

Версия 0.0.1 **Сайт** www.bitswash.org

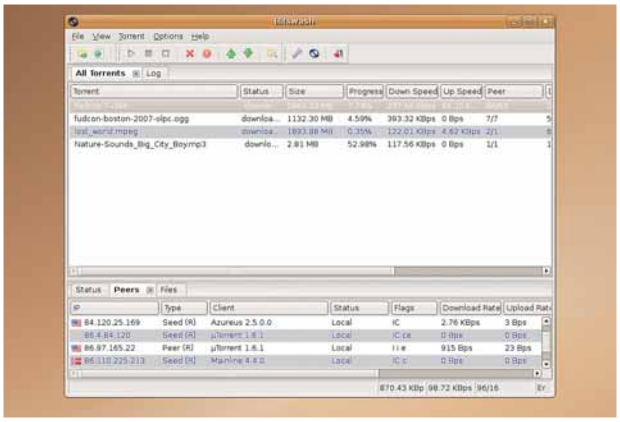
Открытый исходный код делает серьезные заявки, и *Bitswash* вам это докажет. Взгляните на номер версии: всего лишь 0.0.1. Тем не менее *Bitswash* – сказочно быстрый BitTorrent-клиент со множеством функций прямо из коробки. Чтобы достичь такого уровня в первом релизе, *Bitswash* был построен на базе wxWidgets (*libwxgtk2.8 -dev*, если у вас Ubuntu), а также библиотек C++ Boost (в частности, *DateTime, Fileszstem, Serialiyation, Thread* и *Regex*), но других зависимостей нет. Имея все это, вам нужно лишь выполнить *./configure*, *make* и *make install*, в чем вы не раскаетесь: *Bitswash* требует очень мало системных ресурсов и обладает всеми преимуществами многих других клиентов BitTorrent.

Легкий и красивый

Bitswash показывает, сколько всего можно сделать, не пожирая сотен мегабайт ОЗУ. Для добавления торрента из URL достаточно нажать Ctrl+U, затем ввести URL; он сразу же оперативно отслеживается, и начинается загрузка. Используя официальный клиент BitTorrent, я часто сталкивался с проблемами – он не дружил с моим брандмауэром; а *Bitswash* автоматически выбирает нужные порты, и сроду не обеспокоил меня сообщениями. Интерфейс пользователя чрезвычайно прост: в верхней половине – информация о прогрессе закачки, нижняя – разбита на вкладки с информацией. Я думаю, вам больше всего понравится вкладка Peers, где *Bitswash* автоматически распознает географию IP-адресов и, где возможно, выставляет миниатюрные флажки стран, что довольно мило и показывает, как разросся охват BitTorrent.

Как и полагается нормальному инструменты обмена файлами, *Bitswash* позволяет оценивать скорость скачки и закачки в Kbps; при желании обеспечить справедливое распределение можно даже выставить процентовку. Однако есть пара подводных камней: рассчитывайте найти ссылки на Bitwave, не ожидайте, что заработает функция минимизации в системный лоток, и, безусловно, будьте готовы к эпизодическим зависаниям программы; но это не особая проблема, так как после перезапуска приложения все пересылки продолжатся с того же места.

> «Bitswash распознает страну IP-адреса и показывает ее флаг.»



› **Управление вашими закачками под Linux никогда не было столь простым.**

И ведь это только начало – это самый первый альфа-релиз, а он уже выглядит круто. Пока *Bitswash* остается быстрым и легким, его ждет яркое будущее – испытайте его!

Сканер

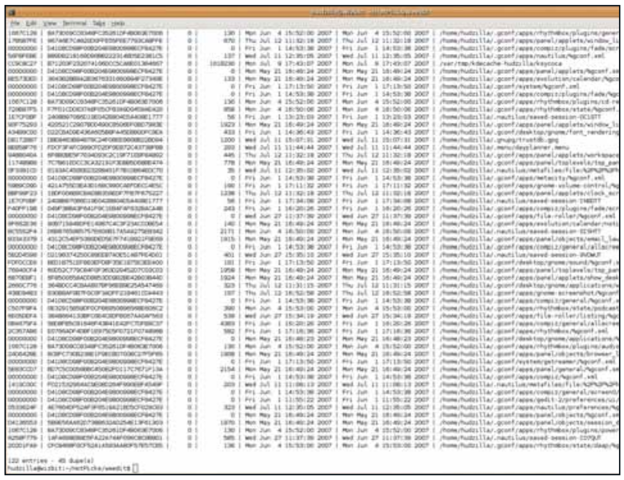
WeedIt

Версия 2.0 **Сайт** http://adm1n.cjb.net/cw

Великолепная программа, помогающая уменьшить захламленность жесткого диска: она сканирует папки на наличие дубликатов файлов, а затем сохраняет все свои выводы в базу данных для последующей проверки.

Для каждого обнаруженного файла *WeedIt* подсчитывает хэш-сумму MD5, по простому математическому алгоритму, генерирующему уникальные (с высокой вероятностью) идентификаторы данных. Эта хэш-сумма затем сохраняется и сравнивается с другими найденными *WeedIt* хэш-суммами – именно так определяются дубликаты. Иногда бывают ложные тревоги: *WeedIt*, похоже, не слишком хорошо справляется с символическими ссылками, так что не спешите удалять все, что она объявит лишним! Чтобы установить *WeedIt*, извлеките ее из архива и запустите *make* в каталоге *WeedIt*. Для работы программы нужно указать ей каталог, где сканировать; команда *weedit ~* просканирует ваш домашний каталог. При каждом нахождении дубликатов файлов они будут добавляться в кэш, и программа не будет возвращаться к ним снова. Если вы хотите начать с нуля, просто удалить файл кэша **weedit.dat**.

WeedIt не особо сложен; если вы столкнетесь с проблемой, попробуйте *readdb* (идет в комплекте с *WeedIt*). Чтобы собрать эту утилиту, перейдите в каталог **utils** и запустите команду *gcc -o readdb readdb.c*. Затем запустите *readdb ./weedit.dat* (указав, где находится ваш файл .dat), и *readdb* распечатает все файлы в кэше, а также их суммы MD5.



› Не получается с определением повторяющихся файлов? Попробуйте readdb, утилиту *WeedIt*, и поймете, в чем проблема.

Календарь-планировщик

Day Planner

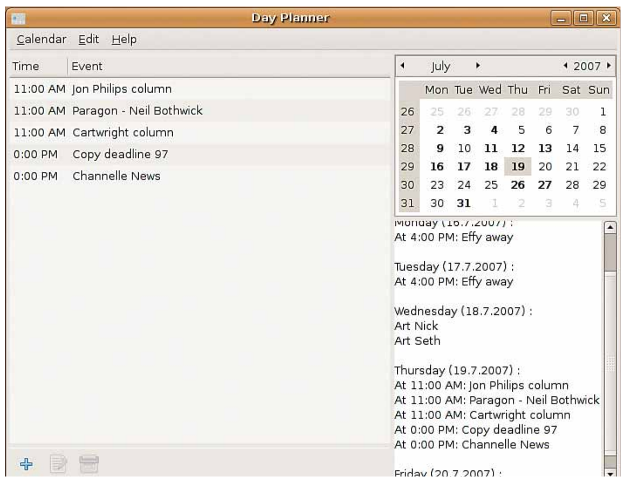
Версия 0.7.1 **Сайт** www.day-planner.org

В отличие от больших приложений типа *Evolution*, *Day Planner* не делает попыток отслеживать сообщения, интегрироваться с вашим IM-клиентом, читать RSS-каналы и т.д. – это простой небольшой календарь с малым количеством зависимостей, с ходу понятный всем знакомым с основными интерфейсами Gnome.

Day Planner поставляется как уже скомпилированный .run-файл. Запустите **sh dayplanner-0.7.1.run**, но не как суперпользователь: *Day Planner* автоматически устанавливается в домашний каталог, и если вы будете устанавливать его из-под **root**, то только root будет иметь к нему доступ! После установки должен появиться ярлык на рабочем столе, но у нас этого ни разу не получилось. Вместо этого, найдите папку **.local** в домашнем каталоге (по умолчанию она скрыта), а затем перейдите к **.local/share/dayplanner** и запустите *dayplanner*.

При первом запуске событий в *Day Planner* не содержится; введите их через меню Edit > Add An Event или Ctrl+A. *Day Planner* позволяет задать тип добавляемого события. Запустив *Day Planner*, вы будете автоматически оповещаться о предстоящих событиях и сможете экспортировать свой календарь в ICS-файл, который легко синхронизируется с онлайн-календарями и даже с вашим телефоном, если он поддерживает этот стандарт. Вот и все: никаких лишних наворотов. LXF

› Как и все хорошие приложения под Unix, *Day Planner* делает одну простую задачу, и делает очень хорошо.



Также выпущены

Новые и обновлённые программы, заслуживающие вашего внимания...

shUnit 2.0.3 Unit-тесты для оболочек типа *Bash* http://forestent.com/projects/shunit2

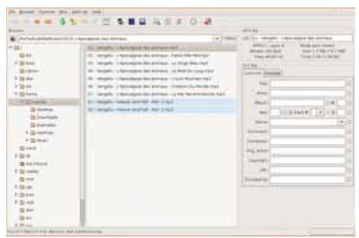
Ultimate++ 707-dev1 Кросс-платформенный RAD для C++ www.ultimatepp.org

gtkpod 0.99.10 Поддержка iPod для GTK www.gtkpod.org

Mesk 0.3.2 Аудиоплейер на языке Python http://mesk.nicfit.net

Taffy 0.57 Увлекательный язык скриптов, по типу Ruby http://freshmeat.net/projects/taffy-lang

EasyTAG 2.1.2 Легкий в использовании редактор метаданных для музыкальных файлов http://easytag.sourceforge.net



› Интерфейс *EasyTAG* требует привычки, но содержит множество информации.

pretendroot 0.9 Одурачьте вашу файловую систему: пусть она думает, что вы суперпользователь ftp://ftp.uhulinux.hu/pub/sources/pretendroot

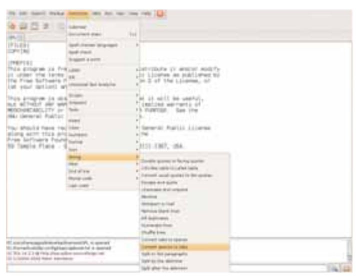
TnFOX 0.87 Инструментарий GUI для C++ и Python www.nedprod.com/TnFOX

Bacula 2.1.26 Резервное копирование, сохранение, восстановление – в одной посуде www.bacula.org

MyRPM 4.36 Создание RPM без хлопот http://code.google.com/p/myrpm

Crossroads Load Balancer 1.51 Балансировка нагрузки для серверов http://crossroads.e-tunity.com

TEA for Linux 17.0.1 Текстовый редактор, с упором на HTML http://tea-editor.sourceforge.net



› *TEA* справляется с разметкой Wikipedia и даже может преобразовать текст в азбуку Морзе!



LXF DVD 97

Не с одним, не с двумя, а с четырьмя дистрибутивами!



Майк Сондерс любовно подбирает содержимое диска Linux Format, а также поддерживает сайт www.linuxformat.co.uk.

Человек-оркестр

В конце 1990-х, когда Linux столкнулся с самыми первыми проявлениями коммерциализации, сообщество Open Source хмурило брови. Не испортится ли Linux из-за бизнеса? Согласятся ли коммерческие предприятия направить эту ОС на рискованные проекты? И нужны ли нам оплачиваемые программисты с полной занятостью? Linux развивался и завоевывал уважение во всем мире; но некоторым разработчикам по-прежнему казалось, что наша ОС должна основываться исключительно на усилиях сообщества.

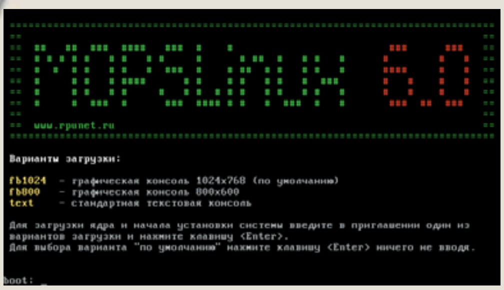
Прошло десять лет, и эти страхи превратились в воспоминания. В целом, такие компании, как Red Hat, Novell и IBM, придали Linux новое ускорение, добавляя новые функции и технологии, чтобы вывести ОС на уровень «продукта корпоративного уровня». Лицензия GPL гарантировала, что никакие безответственные действия не повредят проекту и что никакая компания не станет монопольным поставщиком Linux.

Но лучше всего то, что мелкие проекты дистрибутивов сильны как никогда. Многие полагали, что коммерциализация Linux станет последним звонком для дистрибутивов, создаваемых малыми группами – у них просто не хватит ресурсов и рабочей силы для конкуренции. Однако произошло все в точности наоборот: корпорации-миллионеры плодят новый код для проектов под GPL, а крошечные дистрибутивы в состоянии сами вносить эти изменения.

Вот почему Slackware по-прежнему популярен и силен сегодня. Хотя он почти полностью создан одним-единственным человеком, Патриком Фолькердингом [Patrick Volkerding], все фанаты Slackware могут пожинать урожай от трудов компаний Linux. Если Red Hat добавляет какую-то крутую функцию в *GCC*, для Slackware это тоже выгодно. Это ситуация, когда выигрывают все, и (по крайней мере, на данный момент) сообщество и коммерческий лагерь гармонично сосуществуют. Если, конечно, исключить SCO...

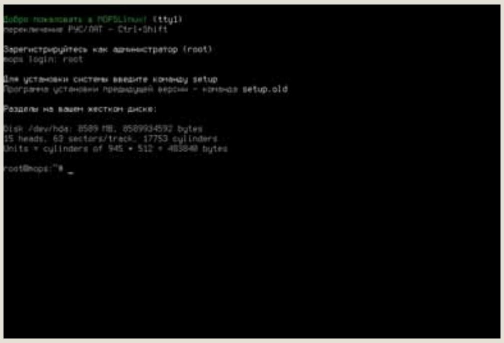
mike.saunders@futurenet.co.uk

Шаг за шагом: Устанавливаем MOPS Linux 6.0



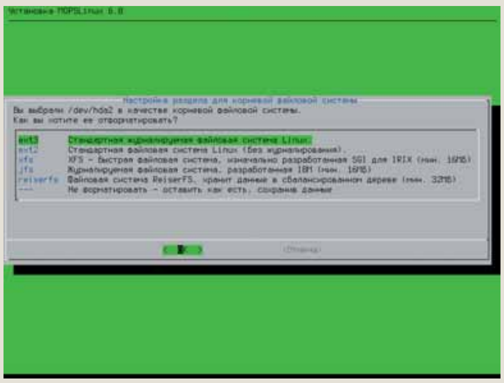
1 Загрузка

Загрузите ваш ПК с **LXF** DVD в приводе, и появится этот экран. Нажмите Enter для загрузки ядра по умолчанию или введите одну из предлагаемых опций.



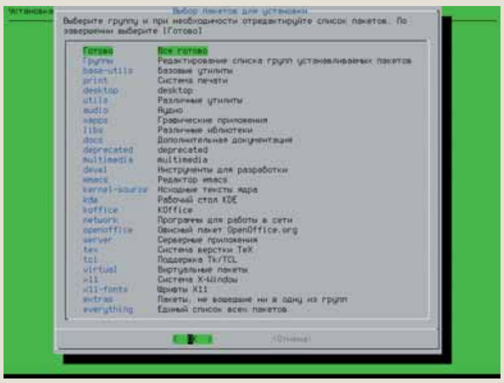
2 Приступаем

Войдите в систему как root и введите setup для запуска программы установки. Если вам больше по душе ее старый вариант, используйте setup.old.



5 Форматирование

Выберите одну из предложенных файловых систем, и программа отформатирует для вас корневой раздел. При необходимости можете пропустить этот этап, выбрав последнюю опцию в списке.



6 Пакеты

Выберите установку с набора CD/DVD, тип инсталляции (ПК, сервер, базовая система или полная инсталляция) и перед вами появится это меню. Затем укажите нужную группу и, при необходимости, отредактируйте список пакетов.

Дистрибутив Linux

MOPSLinux 6.0

MOPSLinux, разработка отечественного НПО «Сеть», базируется на Slackware – одном из самых чистых дистрибутивов в мире. Начиная с самого первого релиза в 1993 году, Slack (как его называют) приобрел верных поклонников, которые ценят его приверженность философии KISS (Keep It Small and Simple – Оставаться Небольшим и Простым), которая действует согласно своему названию: отвергает украшательские навороты, сохраняя структуру дистрибутива простой для понимания.

Поэтому программы для MOPSLinux/Slackware, как правило, идентичны с исходными версиями, т.е. с версиями оригинальных разработчиков (или крайне близки к ним) – в противоположность Ubuntu, Fedora, OpenSUSE и их дружкам, где в пакеты добавляют заплатки ради включения новых функций и подстройки программ к специфике дистрибутива. Конечно, иногда это и полезно, но если вы хотите быть уверены, что получили «чистый» продукт от разработчика, да еще и с очень качественной русификацией и локалью UTF-8, выбирайте MOPSLinux.

Однако считать MOPSLinux русифицированным Slackware было бы ошибкой. Начиная с версии 6.0, в его состав входит собственная система управления пакетами – **mpkg**. В отличие от Slackware, весьма консервативного в данном вопросе, *mpkg* идет в ногу со временем: здесь есть и контроль зависимостей, и их автоматическое разрешение, поддерживается работа со множественными репозиториями – и все это при сохранении совместимости с форматом **.tgz**, принятом в Slackware! Любителей собирать недостающее ПО из исходных текстов порадуют консольные и графические утилиты для создания собственных пакетов.

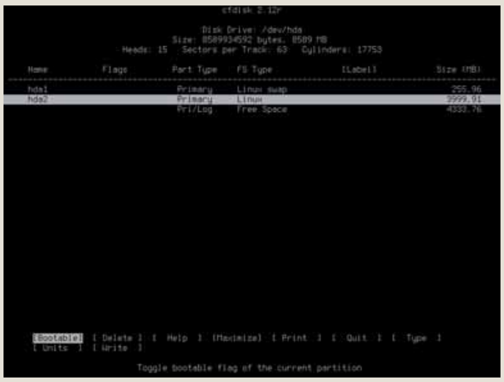
Впрочем, одним *mpkg* дело не ограничивается. Скажите, где вы видели Slackware с трехмерным рабочим столом прямо из коробки? А здесь – пожалуйста: *Beryl 0.2.1* встроен в систему, необходимо только доустановить драйвера к вашей видеокарте из раздела **Главное** нашего DVD.

В отличие от оригинального Slackware, который часто ругают за ограниченную подборку пакетов, MOPSLinux поставляется с куда более богатым набором ПО. В состав версии 6.0, базирующейся на Slackware 12.0, входят: ядро 2.6.22.1, *gcc 4.1.2*, *X.Org 7.3*, KDE 3.5.7, *OpenOffice.org 2.2.1 Professional* (сборка компании Инфра-ресурс), *Lazarus 0.9.22*, *Qt 4.3.1*, *PostgreSQL 8.2.4*, система символьных вычислений *Maxima* с графическим интерфейсом *wxMaxima* и редактором *TeXmacs*, а также различные инструменты для работы с виртуальными частными сетями (VPN): *OpenVPN 2.0.9*, *PPTP Client 1.7.1*, *Kvpnc 0.8.9*.

MOPSLinux – неплохой выбор для старых машин, поскольку его текстовый инсталлятор нетребователен к памяти: чтобы превратить какой-нибудь Pentium-II в небольшой сервер, достаточно 128 МБ ОЗУ и 1 ГБ на диске. Однако для полной установки потребуется 7 ГБ.

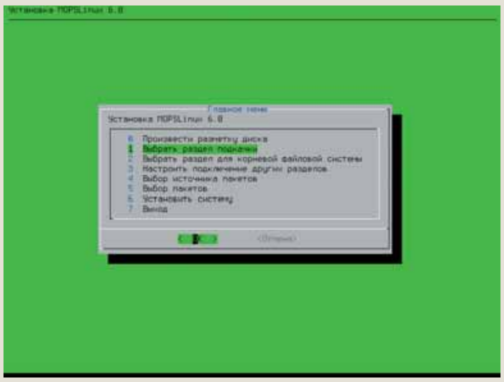


› Экран входа в систему выглядит стильно – но это еще не повод использовать учетную запись root.



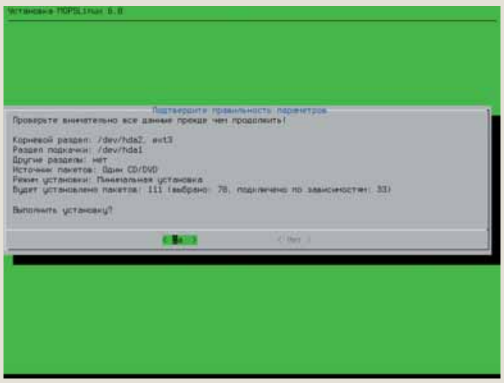
3 Разметка

Используйте стрелки курсора для изменения уже имеющихся разделов и/или создания новых. Вам понадобятся раздел Linux и раздел подкачки. Запишите таблицу разделов на диск и выходите.



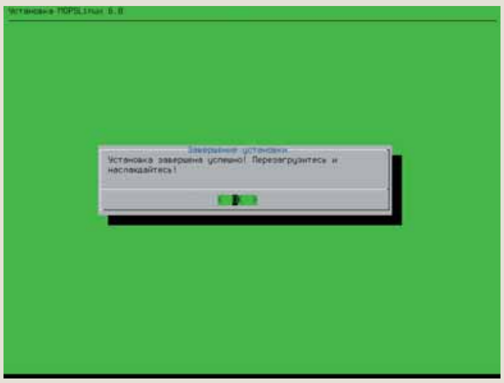
4 Меню

В главном меню инсталлятора выберите пункт под номером 1, а затем укажите раздел подкачки, созданный на предыдущем шаге. Проделайте то же самое и для корневого раздела.



7 Установка

Проверьте правильность введенных данных и, если все выглядит хорошо, смело жмите Да, чтобы начать установку.



8 Окончание

Инсталлятор скопирует файлы и задаст вам еще несколько вопросов по настройке, например, предложит задать пароль администратора. После этого ПК будет перезагружен (не забудьте извлечь DVD) – и готово!

Дистрибутив Linux

Zenwalk 4.6 Core & Live

Ранее известный как Minislack, Zenwalk (www.zenwalk.org) уже три года находится в стадии серьезной разработки, основываясь на зрелом фундаменте Slackware и помаленьку добавляя графический интерфейс пользователя; на момент написания статьи он занимает достойное уважения 15-е место в рейтинге DistroWatch.

Разработка Zenwalk преследует три основные цели: быть современной, быстрой и рациональной, что очень напоминает основную идею Ubuntu с одной-двумя программами для каждой конкретной задачи. Slackware предоставит вам триллионы опций, а Zenwalk ограничивается одним или двумя яркими примерами каждого программного жанра.

Ключевое слово здесь – производительность: рабочий стол *Xfce* означает значительно меньшую тяжеловесность, чем использование по умолчанию Gnome или KDE. Скомпилированный с оптимизациями для Pentium II, он отлично работает на современном оборудовании, но также совместим и с 486. Системные требования ограничиваются скромными 128 МБ ОЗУ и 2 ГБ на жестком диске.

Имеются четыре редакции Zenwalk: стандартная Standard Edition; Core Edition (базовый дистрибутив с набором консольных приложений; пригодится в качестве фундамента для собственной разработки); версия Live, работающая непосредственно с CD, и серверная Server Edition. Редакции Standard и Server были у нас на диске два месяца назад, а сегодня мы хотим представить вашему вниманию Core и Live.

Итак, Zenwalk – отличный, скоростной настольный дистрибутив с хорошим выбором самого свежего ПО. Рекомендуем его тем, кто хочет приобщиться к подходу Slackware, но ищет нечто попроще, а также тем, у кого есть безработные старые машины, которым не по силам тяжесть Fedora.

Zenwalk представлен на нашем диске в форме ISO-образов. Зайдите в раздел **Дистрибутивы/Zenwalk** нашего DVD, и вы увидите там файлы с названиями **zenwalk-core-4.6.iso** и **zenwalk-live-4.6.iso** – это и есть образы дисков, которые нужно записать в загружаемом формате. Копировать файлы на CD не надо; запишите их именно как образы. Например, при работе в *K3b* зайдите в меню Tools и выберите ‘Burn CD Image...’

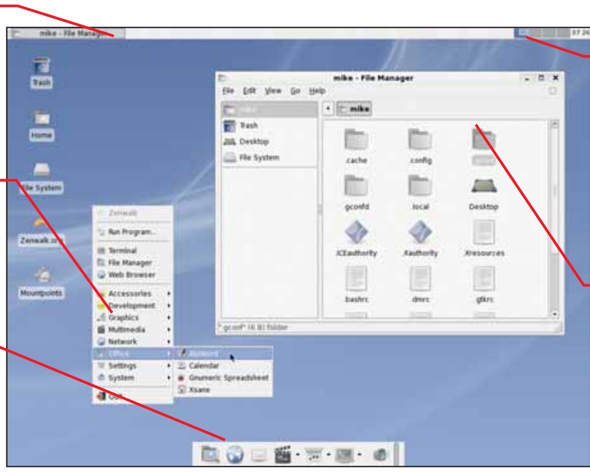
После загрузки Zenwalk Live вы увидите рабочий стол *Xfce*, основные элементы которого показаны на рисунке вверху. По умолчанию вы входите в систему как пользователь one; если вам зачем-либо потребуются права root, используйте пароль 'ZenLive' (регистр важен). Дистрибутив поддерживает множество загрузочных параметров (cheat codes) – подробности можно найти в руководстве пользователя, ярлык которого (User Guide) расположен прямо на рабочем столе. Например, чтобы заставить Zenwalk говорить по-русски, достаточно нажать клавишу Tab при появлении экрана загрузчика и изменить lang=en_US на lang=ru_RU.

Исследуем рабочий стол Zenwalk

Панель
Здесь находится панель задач, где можно переключаться с одной работающей программы на другую.

Меню
Щелкните правой кнопкой мыши по рабочему столу, чтобы вызвать меню программ Zenwalk.

Док
На этой панели можно размещать значки запуска программ и небольшие апплеты.



Рабочее пространство
Нажмите любой из четырех квадратиков – и перейдете на другой виртуальный рабочий стол.

Thunar
Это – файловый менеджер *Xfce*, заменивший *Xffm*, который использовался в прошлых релизах.

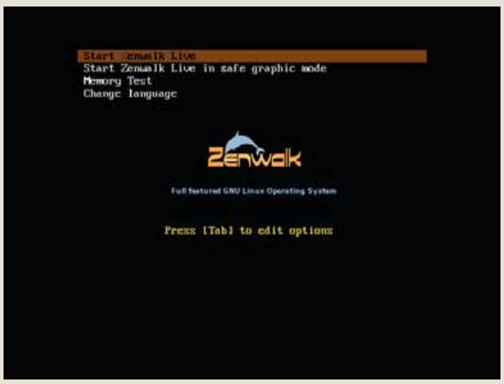


› ZenLive включает в себя подробное руководство пользователя.

Шаг за шагом: Запускаем Zenwalk 4.6 Live



1 Прожиг диска
Запишите ISO-образ Zenwalk Live раздела Дистрибутивы на CD-R. Для этого можно использовать *K3b* (в KDE) или GnomeBaker (в Gnome). Если вы умеете работать с командной строкой, используйте *cdrecord*.



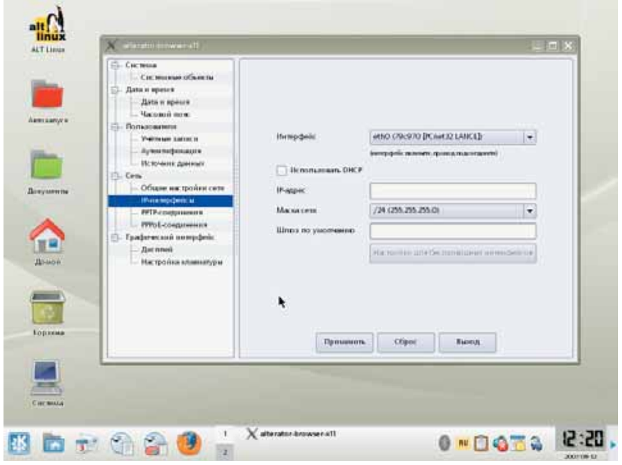
2 Загрузка
Возьмите свежезаписанный CD-R и загрузите компьютер с него. Опции загрузки по умолчанию должны быть нормальными, но в случае проблем, нажмите F2 на экране загрузки, чтобы видеть подробности происходящего; полезные советы найдутся на форумах http://support.zenwalk.org.

Дистрибутив Linux

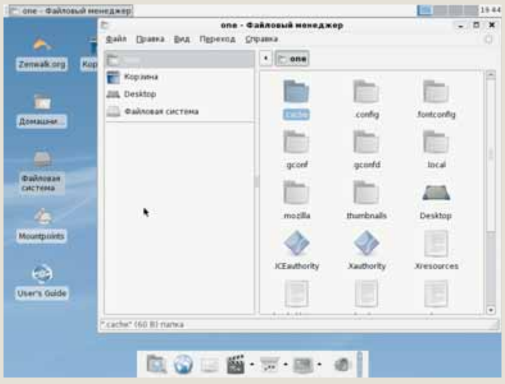
ALT Linux 4.0 Personal Desktop

Бывают дни, когда наш редакционный ящик просто раздувается от писем с предложениями «выложить ALT на диске ближайшего номера». Спешим обрадовать наших респондентов – ваши просьбы были нами услышаны, и на второй стороне DVD этого месяца с комфортом разместился ALT Linux 4.0 Personal Desktop. Этот отечественный дистрибутив предназначен для офисного и домашнего применения, поэтому разработчики постарались предоставить пользователю возможность решать обычные задачи в удобном окружении. Операционная система укомплектована пакетом офисных приложений *OpenOffice.org*, позволяющим работать с документами как в открытом формате ODF, так и в формате *Microsoft Office*. Широко представлены средства работы в Интернете, программы для работы с векторной и растровой графикой, мультимедиа, словари и обучающие программы. Для запуска Windows-приложений в среде Linux предлагается пакет *Wine*. Дистрибутив отличается невысокими требованиями к оборудованию и обладает полноценной локализацией, значительно облегчающей освоение операционной системы пользователями с малым опытом работы на компьютере. В числе новинок можно отметить появление программы *ALTerator*, позволяющей осуществлять управление системой и изменять большинство популярных настроек в удобном графическом режиме. Также впервые на одном DVD-носителе объединилось сразу три версии данного программного продукта: помимо установочного комплекта на здесь представлен LiveCD и восстановительный диск. Выбор между вариантами запуска осуществляется с помощью графического меню. Установка ALT Linux 4.0 Personal Desktop возможна не только на выделенный компьютер, но и в качестве второй операционной системы на свободное место раздела, не занятое файлами операционной системы Microsoft Windows.

Для того, чтобы пользователь не запутался в массе однотипных приложений, в данную операционную систему собраны только наиболее популярные пакеты, благодаря чему основное меню не перегружено и позволяет легко ориентироваться среди разделов и программ. Оформление рабочего стола выполнено в спокойном стиле; отдельное спасибо можно сказать разработчикам за качественную подборку шрифтов. Для управления пакетами предусмотрена программа *Synaptic*, автоматически разрешающая зависимости. Отличительной чертой ALT Linux 4.0 Personal Desktop является наличие проприетарных видеодрайверов, подключаемых уже на стадии установки операционной системы, а также почти всех необходимых кодеков для воспроизведения музыки и видео.



› *ALTerator* позволяет легко настроить основные параметры системы.



3 Работа

Подождите, пока запустится рабочий стол Xfce и наслаждайтесь комфортной рабочей средой. Меню для запуска приложений доступно по правому щелчку мышью.

Документация
Статьи в формате PDF

В этом месяце предлагаем вашему вниманию впечатляющую коллекцию **Сравнений** из предыдущих выпусков **LXF**, она поможет вам подобрать самые подходящие программы для работы и отдыха. Зайдите в раздел Журнал/PDF на DVD и загрузите **index.html**, где вы увидите 10 полных статей в формате PDF* – их можно прочесть, используя *Kpdf* в KDE или *Evince* в Gnome. (Если у вас не установлена программа для чтения PDF, поищите в менеджере пакетов *Xpdf* – это небольшая и быстрая программка для чтения, имеющаяся в каждом дистрибутиве.)

Для начала, web-браузеры: программы, приобщающие или отвращающие вас от Интернета. Многие из нас, единожды выбрав себе браузер, уже не меняют его, но ведь это – быстро развивающаяся область, и всегда есть смысл познакомиться с другими проектами. Кроме того, хотя некоторые браузеры, например, *Firefox* и *Konqueror*, отлично работают на современных машинах, иногда нужно что-то менее требовательное (например, если вы заходите на сервер без X). Поэтому в нашем **Сравнении** нашлось место и для парочки текстовых браузеров.

Если вы программист, не пропустите обзоров IDE Java и компиляторов C/C++. В первом мы рассматриваем пакеты-гиганты, типа *Eclipse* и *JBuilder*, сравнивая свободные IDE с их коммерческими, закрытыми конкурентами. Наше Сравнение *C/C++* – один из самых глубоких в техническом отношении материалов: Бьяджо Лучини [Biagio Lucini] подвергает *GCC*, *ICC*, *Tendra* и прочих бесчисленным тестам. *GCC* является фактическим стандартом компилятора для GNU/Linux, но означает ли это, что он – лучший?

Кроме того, у нас есть скрупулезные **LXF**-Сравнения файловых менеджеров, движков для блогов, DVD-плейеров, Live-дистрибутивов и X-терминалов. Если вы любитель звездного неба, не пропустите наш тест астрономических программ с открытым кодом, среди которых потрясающий исследователь вселенной *Celestia*.



› Надоел Freshmeat ? Наши PDF помогут подобрать приложения получше.

* Частично на английском языке.

Прочее ПО

Новые релизы

После Ubuntu, вынесенного на обложку журнала, поклонники KDE могут почувствовать себя обделенными. Но не бойтесь: у вас тоже есть возможность заглянуть в будущее через второй промежуточный релиз грядущего KDE 4. На DVD прошлого месяца был Live-дистрибутив, загружавший KDE 4 Beta 1 – в нем хватало ошибок, и все же он давал неплохие шансы оценить, что нас ждет. В этом месяце мы предлагаем полный исходный код ко второму бета-релизу, наглядно показывающий, насколько разработчики KDE продвинулись вперед.

Beta 2 распространяется в виде исходных текстов, так что вам придется самостоятельно собрать для него двоичные пакеты. Это довольно суровая задача; если раньше вы этим не занимались, возможно, стоит сперва потренироваться на программах поменьше с нашего диска (например, с *XChat* в разделе **Интернет**). Освоившись с компиляцией исходного кода, можете заняться и KDE 4, следуя руководству **build_instructions.html**, которое поставляется вместе с tar-архивом. Естественно, для этого вам понадобятся соответствующие инструменты – точные наименования зависят от дистрибутива, но вам потребуется компонент *G++* из *GCC* вместе с библиотекой и заголовочными файлами *Qt* (см. раздел **Разработка**). Найдите в менеджере пакетов *GCC* и установите также и его зависимости; затем следуйте руководству, при необходимости цепляя недостающие зависимости.

Самое потрясающее обновление в KDE 4 – это *Plasma*, новая оболочка рабочего стола, активно использующая апплеты. Подобно *Dashboard Widgets* из OS X и Vista *Gadgets* из Windows, *Plasmo'иды* KDE 4 – это небольшие программки, которые можно скинуть на рабочий стол, чтобы они показывали, например, прогноз погоды, биржевые котировки или ссылки на словари. Пользователи KDE, которые раньше баловались с *SuperKaramba*, найдут в них довольно большое сходство. Еще одно крупное изменение – интеграция Системных Настроек [System Settings], заменивших старый добрый Центр Управления KDE [Control Center]. Пока они выглядят несколько голыми, но по завершении работ навигация в них будет легче (будем надеяться), чем в *KControl*.

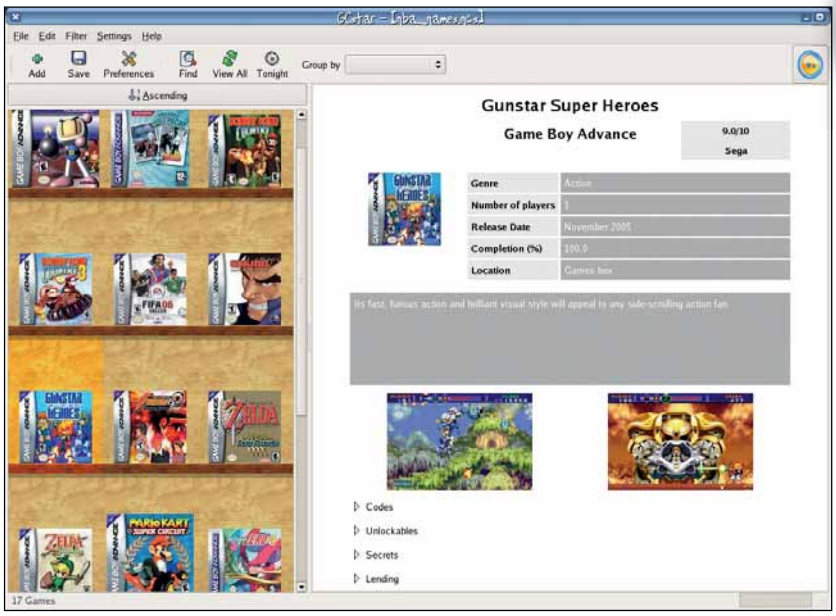
Итак, если хотите увидеть все эти новшества в действии, ныряйте в код и компилируйте. Рекомендуем запастись парочкой гигов свободного места на диске, поскольку в процессе компиляции будут создаваться временные файлы, и было бы неплохо создать вторую учетную запись пользователя, как описано в руководстве. Если вы этого не сделаете, настройки KDE 4 могут затереть ваши собственные настройки KDE, и когда вы снова вернетесь к KDE 3, начнется путаница. В зависимости от скорости вашей машины, компиляция может занять несколько часов – но вам незачем надзирать за всем процессом. Можете выпить кофейку!

Рождение звезды

Краткий обзор прочих интересных программ на диске: *GCStar* в разделе **Рабочий стол** – превосходный менеджер коллекций, он поможет вам организовать ваши видео, музыку, CD, книги и т.п. Вообще говоря, он годится практически для любого вида коллекции. Если вы собираете, допустим, крышечки от йогурта, и *GCStar* вам помог, пришлите снимок экрана на lxf.letters@futurenet.co.uk, самые забавные мы опубликуем в следующем выпуске! Программа написана на Perl, и вы можете установить ее, распаковав файл **gcstar-1.2.2.tar.gz** и запустив скрипт 'install' (от имени суперпользователя) в получившейся директории.

Если вас одолел спам, зайдите в раздел **Сервер** поглядеть на *Mail Avenger*, почтовый сервер, гоняющий ваши сообщения по всевозможным фильтрам, чтобы вычистить мусор. В сочетании с *ClamAV*, он также может удалять вирусы из электронной почты – для линуксоидов это, может, и не очень актуально, но все же полезно, если ваш почтовый сервер имеет дело с пользователями Windows. Также в разделе **Сервер** есть *Bftpd*, полезный небольшой FTP-сервер, почти не нуждающийся в настройке. Если вам надо скинуть файлы с одной машины на другую, но неохота тратить вечность на настройку SSHD для SCP, эта программа легко и быстро вас выручит.

И, наконец, все программисты должны познакомиться с *Highlight* в разделе **Разработка**. Эта полезная утилита конвертирует исходный код аж из 120 языков в красочные страницы HTML, и даже имеет поддержку Ассемблера x86. О большем и мечтать не приходится!



› Организуйте свою коллекцию игр/книг/видео/марок/бабочек с помощью *GCStar*.

И напоследок...



Можно отлично развлечься с помощью *SuperTuxKart* в нашем разделе **Игры**. Это расширенная версия *TuxKart* (вдохновленная *Super Mario Kart*) с целой галереей персонажей, новыми игроками, трассами и программами AI – выглядит весьма презентабельно. Чтобы начать игру, скопируйте файл **.tar.bz** из директории игры в домашнюю директорию, затем откройте терминал и введите:

```
tar xfvj supertuxkart-0.3-linuxi486.tar.bz2
cd supertuxkart-0.3-linuxi486/
./run_game.sh
```

В открывшейся директории прочитайте файл README, это инструкция по управлению. Поклонникам ролевых игр старой школы времен NES предлагается *Monster* – говорят, что на игровой процесс здесь уходит 5–10 часов от начала до конца. А если вы любите *The Beautiful Game*, дайте шанс *Bygfoot* – это имитатор футбола, у него как раз вышел релиз 2.2. LXF



› Проложите путь, огибая банановые кожурки Doom™ в *SuperTuxKart*.

Информация о диске

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



Содержание LINUX FORMAT DVD Октябрь 2007

Сторона 1

Рабочий стол
Ananas - система автоматизации учета
GCstar - приложение для управления коллекциями
GnuCash - менеджер финансов
K3b - программа для записи CD/DVD-дисков
KDE - исходные коды второй бета-версии KDE4
Kino - нелинейный DV-редактор
MuseScore - WYSIWYG-редактор нот
Parcelle - утилита для управления буфером обмена
pyRenamer - программа для массового переименования файлов
Rubrica - адресная книга
Tea - текстовый редактор
themonospot - утилита для определения кодеков видеофайла

Разработка
EyeD3 - модуль для работы с ID3-тегами на Python
Gaphor - простой в использовании UML-редактор
GtkSourceView - компонент редактирования текста
Highlight - конвертер исходников в различные форматы
Mono - свободная реализация .NET
MonoDevelop - среда разработки для Mono
Qt - инструментарий для написания кроссплатформенных приложений

Дистрибутивы
MOPSLinux - русскоязычный дистрибутив, основанный на Slackware
Zenwalk - облегченная десктопная версия Slackware

Игры
Bygfoot - футбольный менеджер
Monster - ролевая игра в стиле RPG для NES/SNES
Stax - набор игр с падающими блоками
SuperTuxKart - улучшенная версия TuxKart
TuxMathScrabble - математическая игра

Справка
Ответы на часто задаваемые вопросы
RUTE - книга по системному администрированию Linux

Hot Pics
Bitswash - кроссплатформенный клиент BitTorrent
Day planner - органайзер
Herria - музыкальный проигрыватель для командной строки
JSCoverage - генератор статистики для кода на JavaScript
Segatex - графический интерфейс для настройки параметров SELinux
Slingshot - двумерная пошаговая стратегическая игра
Thunder&Lightning - футуристический трехмерный авиасимулятор
WeedIt - утилита для нахождения дубликатов
Wiki on a Stick (WoaS) - wiki в самомодифицирующемся XHTML-файле
Zero Install - менеджер пакетов

Интернет
ChatZilla - кроссплатформенный IRC-клиент
Irssi - IRC-клиент с интерфейсом командной строки
Konversation - IRC-клиент для KDE
Kopete - клиент мгновенных сообщений
KVIrc - свободный IRC-клиент на Qt
Lightning - календарь для Mozilla Thunderbird
Pidgin - кроссплатформенный AIM/ICQ/MSN/Jabber-клиент
Provider - расширение для работы Sunbird и Lightning с Google Calendar
SeaMonkey - набор Интернет-приложений
Sunbird - календарь-органайзер
Sylpheed - легковесный почтовый клиент
XChat - IRC-клиент для Linux и Windows

Сервер
Bftpd - гибкий FTP-сервер
ClamAV - свободный антивирус
Dovecot - IMAP/POP3-сервер
GTKO - утилита разработки и тестирования для Oracle DB
Mail Avenger - SMTP-демон

Система
Grsync - графический интерфейс для rsync
memtester - утилита для тестирования памяти на ошибки
Rootkit Hunter - сканер руткитов
Zsh - продвинутая интерактивная оболочка

Сторона 2

Alt Linux Personal Desktop 4

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен на Уральском электронном заводе., 620066, Россия, г. Екатеринбург, ул. Комвузовская 17-203, Лицензия МПТР России ВАФ № 77-15

Поставляется вместе с журналом LINUXFORMAT номер 10(97) Октябрь 2007

Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Содержание DVD

ЖУРНАЛ
Answers — Файлы из раздела Ответы.
Mono — Код примеров статьи.
Java EE — Файлы к уроку Java.
PDFs — Статьи предыдущих LXF
Python — Код из урока по Python Ника Вейча.

РАБОЧИЙ СТОЛ
GCstar — Менеджер коллекций.
GnuCash — Персональный финансовый менеджер.
K3b — KDE-приложение для прожига CD/DVD.
KDE — Среда рабочего стола.
Kino — Цифровой редактор видео.
MuseScore — Программа записи музыки для KDE (см. картинку).
Parcelle — Менеджер буфера обмена.
PyRenamer — Программа массового переименования файлов.
Rubrica — Адресная книга.
TEA — Супермощный текстовый редактор.
themonospot — Анализатор видеофайлов.
Ананас — Свободная учетная платформа.

РАЗРАБОТКА
eyeD3 — Модуль Python для обработки тэгов ID3.
Gaphor — Инструмент моделирования UML.
Highlight — Программа форматирования исходного кода.
Mono — Открытая реализация .NET
MonoDevelop — Интегрированная среда разработки C#.
Qt — Среда разработки приложений.

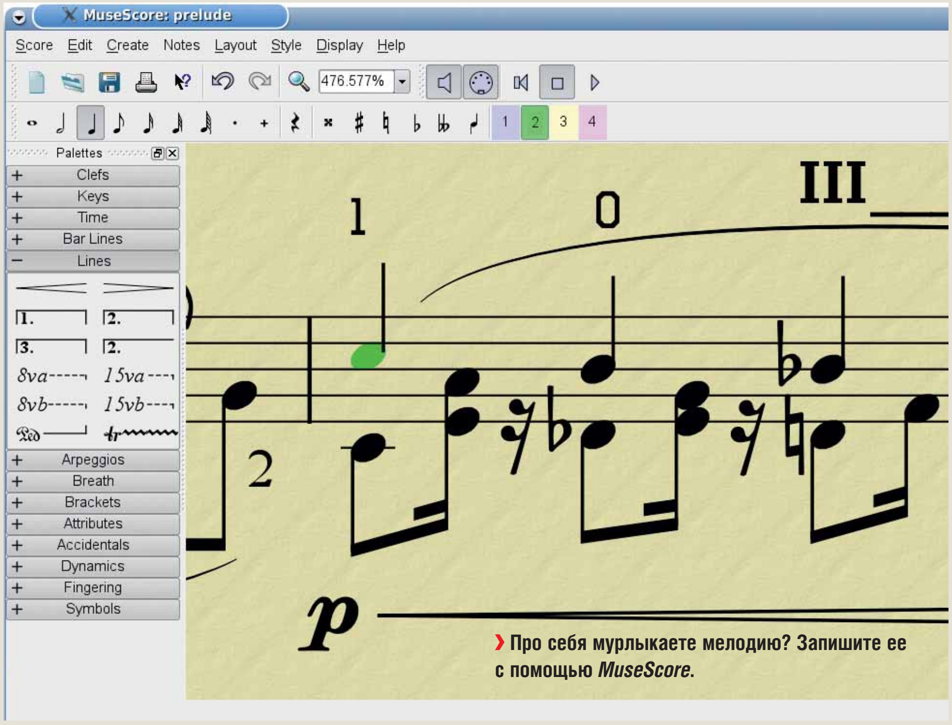
ДИСТРИБУТИВЫ
ALT Linux — Настольная ОС от российского производителя (сторона 2)
MOPS Linux — Отечественный дистрибутив на базе Slackware.
Zenwalk Live — Дружелюбная к пользователю система на CD.

ИГРЫ
Bygfoot — Имитатор футбола.
Monster — RPG в классическом стиле.
Stax — Головоломка с падающими блоками.
SuperTuxKart — Скоростные гонки клювоносых.
TuxMathScrabble — Обучающая игра для детей.

СПРАВКА
Answers — Решенные проблемы Linux.
RUTE — Руководство по администрированию Linux.



› **Антиквариат: ролевая игра старой школы в *Monster*.**



› **Про себя мурлыкаете мелодию? Запишите ее с помощью *MuseScore*.**

HOTPICKS
Bitswash — Кросс-платформенный клиент BitTorrent.
Day Planner — Менеджер календаря.
Herrie — Музыкальный плейер командной строки.
JSCoverage — Программа проверки работы JavaScript.
Segatex — Инструмент SELinux.
Slingshot — Космическая стрелялка.
Thunder and Lightning — Трехмерный имитатор войнушки.
Weedlt — Сканер файлов-дубликатов.
Wiki on a Stick — Wiki в вашем браузере.
Zero Install Injector — Децентрализованный менеджер пакетов.

СИСТЕМА
Grsync — Графический интерфейс для rsync.
Klogshow — Монитор файлов журналов.
memtester — Программа тестирования памяти.
RKHunter — Программа поиска руткитов.
Zsh — Мощная оболочка.

ИНТЕРНЕТ
ChatZilla — Основанный на XUL клиент IRC.
Irssi — Текстовый клиент IRC.
Konversation — Клиент IRC для KDE.
Kopete — Интернет-пейджер KDE.
KVIrc — Клиент IRC на базе Qt.
Lightning — Календарь Thunderbird.
Pidgin — Мультипротокольный пейджер.
Provider — Связь с календарем Google.
SeaMonkey — Интернет-пакет.
Sunbird — Календарь Mozilla.
Sylpheed — Нетребовательный к ресурсам почтовый клиент.
XChat — Основанный на Gtk клиент IRC.

СЕРВЕР
Bftpd — Быстрый FTP-сервер.
ClamAV — Антивирусный набор инструментов.
Dovecot — Надежный IMAP-сервер.
GTKO — Интерфейс базы данных Oracle.
Mail Avenger — SMTP-демон против спама.

ГЛАВНОЕ
ATI driver — Графический драйвер.
Bash — Оболочка командной строки.
Cairo — Библиотека двумерной графики.
CheckInstall — Программа создания двоичных пакетов.
Coreutils — Утилиты командной строки.
CSV — Индекс файлов диска.
GLib — Низкоуровневая библиотека.
Glibc — Библиотека GNU C.
Gtk — Инструментарий пользовательского интерфейса.
HardInfo — Системная информация и сравнительный анализ.
Jigdo — Создатель ISO-образов.
Kernel — Свежий релиз ядра Linux.
LibXML2 — XML-анализатор и инструментарий.
ncurses — Оконный инструментарий текстового режима.
Nvidia driver — Графический драйвер.
Python — Язык программирования.
RAWRITE — Запись образов на дискеты.
SBM — The Smart Boot Manager.
SDL — Библиотека мультимедиа.

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 6000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Елена Толстякова, Иван Мищенко
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев, Татьяна Цыганова, Юлия Шабунио
Редактор диска
Александр Кузьменков
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____

Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч [Nick Veitch] nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон [Paul Hudson] paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс [Mike Saunders] mike.saunders@futurenet.co.uk
Редактор обзоров
Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.co.uk
Литературный редактор
Мэтт Нейлон [Matt Nailon] mnailon@futurenet.co.uk
Ассистент по выпуску
Шарлотта Мартин [Charlotte Martyn], Сара Гейнз [Sarah Gaines]

Подготовка материалов
Ладислав Боднар [Ladislav Bodnar], Нейл Ботвик [Neil Bothwick], Д-р Крис Браун [Dr. Chris Brown], Энди Ченнел [Andy Channelle], Крис Хауэлл [Chris Howells], Энди Хадсон [Andy Hudson], Марко Фиоретти [Marco Fioretti], Йон Филлипс [Jon Phillips], Рэйчел Проберт [Rachel Probert], Евгений Балдин, Александр Бабаев, Андрей Боровский, Михаил Журухин, Владимир Иванов, Андрей Паскаль, Галина Пожарина, Александр Поносов, Петр Семилетов, Сергей Супрунов, Алексей Федорчук, Виктор Федосеев, Александр Чернышов, Илья Шпаньков, Сергей Яремчук
Художественные ассистенты: Ник Кокс [Nick Cox], Дэвид Диас [David Dyas], Дебби Воссер [Debbie Vosser], Сетнам Сингх [Sethnam Singh], Зиги Бейкер [Zigi Baker], Майк Бреннан [Mike Brennan]
Фотографии: Ник Вейч [Nick Veitch]
Иллюстрации: Нейл Бартлетт [Neil Bartlett], Пол Блечфорд [Paul Blachford], Крис Винн [Chris Winn]

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург: ул. Гончарная, 23, офис 54, телефон: +7 (812) 717-00-37
Email: info@linuxformat.ru, Web: www.linuxformat.ru
Представительство в Москве:
г. Москва, ул.Энергетическая, д. 14, корпус 5, строение 1.
По вопросам сотрудничества, партнерства, оптовых закупок: +7 (495) 799-18-63
Дирекция московского офиса: +7 (495) 136-88-45
E-mail: moscow@linuxcenter.ru



в ноябре

Summer of Code 2007

Очередной летний марафон от Google подошел к концу – самая пора оценить, насколько успешным он вышел.

KDE 4: мечта становится явью

Самый горячий свободный проект приближается к релизу – разберемся, что происходит в это время за кулисами.

Держи это в тайне

Почему некоторые компании из мира Open Source придерживают кое-что закрытым.

Сравнение: графические брандмауэры

Настроить межсетевой экран непросто, но есть способ сделать работу легче.

Содержание последующих выпусков может меняться без уведомления